Software Globalisation in Finland:
A State-of-the-practice Survey

Jarkko Immonen and Jorma Sajaniemi

Report A-2003-1

Software Globalisation in Finland:
A State-of-the-practice Survey

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

ABSTRACT

During recent years software globalisation has become an important part of the whole software industry. Adaptation of software to different market-areas is not an easy or straightforward task. The adaptation process usually involves two participants: a software developer and a localisation vendor. We interviewed Finnish software industry representatives in order to find out the current state of globalisation processes and practices and to inquire their views on software globalisation. On the basis of the interviews, it seems that globalisation practices vary greatly and localisation is often considered as purely translation. In addition, there seems to be many problems that could be solved by using more defined processes and by enhancing the communication and co-operation between software developers and localisation vendors.

Keywords: Software globalisation, internationalisation, localisation, software process improvement

1   INTRODUCTION

Today information technology is a part of everyday life in the industrialised world and it is continuously gaining ground in many other countries. Only about a decade ago most computer software was produced in the US for American users by American developers, but nowadays more and more users live outside the traditional software market-areas. This is one of the reasons why software industry has become a global business.

Another reason that has promoted software export is the exponential growth of the Internet which 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 which was 45% of the total software sales (Hietala & al., 2002). The revenues from international business grew by 19%. As tempting as the new markets may seem to be, they also bring new requirements for software developers because a product that is designed for users in one country may not suit users in other countries and cultures as such.

There are millions of people in the world, and they differ from each other not only as individuals but also as representatives of different cultural backgrounds (Hofstede, 1997). There are about 6800 main languages (SIL International, 2002) from which 435 are included in ISO 639-2

standard for commercial use (Harris & McCormack, 2000). Although the fifteen most common languages cover nearly 50% of the world's population, there are more than 100 languages with more than 10 million native speakers (Harris & McCormack, 2000). Beside speaking different languages, people use different kind of writing systems called scripts to represent those languages. There are roughly 30 different living writing systems in the world at the moment (Harris & McCormack, 2000).

In addition to differences in languages and writing systems, people interpret images and symbols in various ways and associate colours with different issues depending on their cultural background. For this reason, user interfaces should be customised for the target group in order to achieve better usability for all kind of users. Some specific notations, such as numbers, currency, time and date, are dependent on the cultural environment of the users as well. These are examples of matters that should be taken into consideration when designing a software product that will be delivered world-wide.

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). DePalma et al. (1998) found that Internet users were three times as likely to buy when addressed in their own language and that the cost of customer service, for example, dropped if the users could read the instructions in their native language. In some cases country specific laws enforce software developers to adapt their products to the target country (O'Donnell, 1994; Vine, 2002; Kokkotos & Spyropoulos, 1997).

Software *globalisation* is the process of developing, manufacturing and marketing software products that are intended for world-wide distribution. This term often refers to marketing context, i.e., a software developer expands marketing from the home markets to the global markets to pursue business opportunities where the customers are located (Esselink, 1998). Globalisation is often abbreviated to G11N (where 11 indicates the number of letters between G and N). Globalisation is a general term which is used to cover two separate processes: internationalisation and localisation. A product is said to be globalised if it is both internationalised and localised (Hall, 2002).

In order to sell a software product world-wide it should be designed and developed in a way that allows it to be translated into other languages and to be adapted to other cultures easily. The most effective way of doing this is to separate culturally dependent elements from culturally independent elements (Taylor, 1992). This process is referred to as *internationalisation* (I18N). During internationalisation all language and culture specific items are isolated and extracted, elements that are considered to be offensive or incorrect to some users are displaced, etc. (Esselink, 1998; Vine, 2002). Internationalisation also covers technical issues, such as supporting different kind of character sets, sorting as well as input and output of international data. This part of I18N is sometimes called *enablement*. An internationalised software product can be translated and adapted to different cultures without making changes to the source code (O'Donnell, 1994). An internationalised software product should be translatable which means that translatable text is not hard-coded within the source code, concatenated strings are not used to create user interface messages, fixed resources are not used, etc. Internationalisation does not require any translation

but it does require that a user can type data in his or her own language and see it displayed properly in that language (Kaplan, 2000). Internationalisation is the task of software developers.

If a software product has not been developed with I18N in mind, it is difficult or even impossible to localise it. *Localisation*, abbreviated to L10N, is the process of translating and adapting software to a particular language and culture (Esselink, 1998). Its target is opposite of internationalisation but similar kind of knowledge about culture, language, social values and expectations is needed (Taylor, 1992). Although translation is usually the main part of localisation, it also includes taking care of local conventions, images and icons, symbols, gestures, colour associations, etc. (Russo & Boor, 1993; Vine, 2002). Unfortunately, due to the high costs of localisation, comprehensive adaptation of a software product is often ignored, and only the user interface and some manuals are translated. There are usually two participants involved in a globalisation process: a software developer and a localisation vendor. In this paper we will use term localisation vendor to cover both localisation vendors and translation agencies with localisation services.

Literature about software G11N has focused more on technical details and general guidelines than the overall globalisation process. For example, practical examples of co-operation between software developers and localisation vendors have been nearly non-existent in the literature. To find out how companies have managed to cope with the situation, we conducted a survey of current globalisation practices and globalisation related problems in Finnish software industry. We interviewed ten Finnish software developers and localisation vendors to get an overview of the software globalisation processes and practices and of the communication and co-operation between 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: what kind of information and skills are needed and what are the most common shortcomings of their employees. This kind of information is useful for improving, for example, translation and computer science curricula in universities.

The rest of the paper is structured in the following way. Section 2 contains a brief glance at the literature related to software G11N and Section 3 concerns the research method used in the survey. The results of the survey are presented in Section 4 with a discussion on the key findings in Section 5. Finally, the paper ends with the conclusions in Section 6.

2 RELATED LITERATURE

Many books and articles related to software globalisation have been published since the early 1990's. This literature can be roughly categorised into two groups as follows. The first group (e.g., Uren & al., 1993; Taylor, 1992; Farid, 1990) consists of papers which describe, for example, how cultural factors should be taken into consideration when producing a global software product and how this kind of software can be implemented in general. The second group (e.g., O'Donnell, 1994; Kano, 1995; Kaplan, 2000; Deitsch & Czarnecki, 2001) consists of books and articles which address more technical issues in detail.

Hofstede (1997), for example, discusses cultural differences and the role of culture in determining characteristics of people. The common part of many cultural models presented in

literature is that they divide culture into a surface level and deeper levels (Hoft, 1996). One often referred article about culture dependent user interfaces has been written by Russo and Boor (1993). They introduce a collection of user interface elements that may not be suitable in other cultures as such. Vine (2002) presents areas that should be taken into consideration while designing user interfaces for an international audience. There are also several articles (Marcus & Gould, 2000; Sheridan, 2001; Zahed & al., 2001) that concern the effects of cultural diversity on user interfaces and web-sites. A book edited by del Galdo and Nielsen (1996) includes articles describing cultural issues in software development more profoundly. According to the book, most software developers take into consideration only the most obvious cultural differences which belong to the surface level of culture.

Internationalisation books are typically twofold because they include sections for general principles and guidelines which should be catered for during I18N as well as sections for detailed technical issues and examples. In addition, the books are often tied to a certain operating system or programming language because they focus mainly on the technical issues of I18N. Uren & al. (1993) describes the fundamentals of software internationalisation and localisation. Different approaches to the production of language variants of a software product have been introduced by Taylor (1992) and Kokkotos & Spyropoulos (1997). Examples of platform- or language-dependent books are O'Donnell (1994) which concentrates on I18N issues using the C programming language in Unix environment and Kano (1995) which describes how to develop international software for Windows environment using the C++ programming language. Kaplan (2000) and Deitsch & Czarnecki (2001) discuss internationalisation using Visual Basic and Java programming languages respectively. Piroumian (1997) gives examples and general knowledge of Java internationalisation. Globalisation of web-sites is discussed, for example, in Yunker (2002).

One of the most famous books about localisation has been written by Esselink (1998). It is directly related to practical localisation, i.e., considers from a translator's point of view how to translate the components of a software product, such as program files, on-line helps and documents, and how to manage localisation projects. Apple Computer Inc. (1992) has published a book about localisation in Macintosh environment. In addition, a few localisation vendors have published their own localisation guides. For example, detailed information about project management with checklists can be found in Rubric's guide (2002) and basic information about the whole localisation process in Lingo System's guide (2000).

As far as we know, there are not many publicly available studies that focus on the processes and practises used in software globalisation at the industry level. Rafii and Perkins (1995) discuss how internationalisation and localisation can be accomplished in parallel cycles in order to reduce the time-to-market. Hall (2002) gives general principles for internationalisation and talks about the position of internationalisation in the software development cycle. Topping shows different approaches to organising localisation (Topping, 1999a) and describes how internationalisation and localisation requirements can be addressed during the development process by using Capability Maturity Model (Topping, 1999b). Nielsen (1996) discusses how the suitability of products can be tested by utilising international user testing methods. Some companies and organisations have published reports, papers and summaries related to this area but often those papers (Global360, 2002; Aberdeen Group, 2001; GlobalSight, 2000; Plumley, 2000; Localisation Industry Standards Association, 2001) are not freely available or they approach the topic not from a technical but, for example, marketing, viewpoint.

3   METHOD

In order to study processes and practices used in the area of software globalisation at the industry level, we conducted a survey in which we interviewed industry representatives from ten Finnish companies.

We selected 23 companies from two groups to be interviewed for the survey. The companies were selected in such a way that they represented different sizes, ages, application domains, etc. Thirteen of the companies were software developers, and ten companies were localisation vendors. The companies were contacted by phone or e-mail. 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 software developers and four localisation vendors were interviewed for the survey. Although the number of the localisation vendors in the survey is smaller than the number of the software developers, their relative proportion is greater than that of software developers as there are very few localisation vendors in Finland.

The method used in the survey was a semi-structured interview. The interviews were based on two structured questionnaire forms, one for localisation vendors and another for software developers. The questions for the questionnaires were mainly formulated on the basis of the G11N literature. Structure of the questionnaire forms and some questions concerning, for example, background of the companies originated from the requirements engineering survey conducted by Nikula & al. (2000). The questionnaire forms can be found in Appendices 1 and 2.

Both questionnaire forms included sections for general information, background information of the company, personnel, projects, technical issues, problems and development needs as well as closing remarks. Some of the questions were similar for both groups but most of the questions were customised for each target group. The questions were mainly multiple choice questions or questions that could be answered with a few words. In addition, there were some open questions concerning facts that needed to be explained more verbosely. Although the interviews were based on the forms, we asked some additional questions to expand upon the subject. All the questions were explained with more details, and the interviewees were given some background information about the topics if it was necessary.

All the interviews were conducted by the first author of this paper. There was one interviewee attending each interview, except in one case in which there were two interviewees. The interviewees acted as management roles in their companies but most of them had additional assignments as well. Six of the interviews took place on the companies' business premises and two of them on a neutral ground. Furthermore, two interviews were conducted using phone.

The interviews were documented on questionnaire forms by the interviewer and they were also recorded on tape. All the tapes were erased after writing down the most important notes. In the case of phone interviews, the accurate notes had to be 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 beforehand by the interviewer.

The interviews lasted between one and two and a half hours with the average of one hour and 45 minutes. The estimated duration for the interviews was about two hours. In the case of shorter interviews, some of the questions were left unanswered due to the lack of the interviewees' time but the missing answers were sent to the interviewer later by post or e-mail.

We encountered some problems during the interviews. Firstly, the questionnaire forms were quite long and extensive. For this reason, there was no time for many additional questions, and in some cases the interviewees had not enough time to discuss all the topics thoroughly. Secondly, two of the interviewees stated that they might not have been the right persons for the interview because they lacked technical or globalisation knowledge. This cannot be seen as a major shortcoming because a part of the questions did not require any special knowledge, and some of the questions could be answered with "not applicable / do not know / no comment" option. Thirdly, some of the companies had just started to globalise their products and thus their processes and methods were unsettled at the moment of the interview.

Since only ten companies were interviewed the results are inadequate for statistical analysis. However, we collected some quantitative data in order to make interpretation and presentation of the results easier. Despite of the small-scale sample, the results of the survey allow us to examine the ways software globalisation is executed in Finnish software companies.

4 RESULTS

This section lists the findings of the survey. Section 4.1 details the companies that participated in the survey and Section 4.2 describes personnel related issues, such as education, skills and in-house training. Section 4.3 focuses on the projects the companies were involved in, and Section 4.4 describes the problems and development needs revealed by the survey and exposes some expectations the interviewees had concerning education and research.

4.1 Companies

The companies in the survey were divided into two groups: software developers and localisation vendors. Six of the interviewed companies were software developers and four of them were localisation vendors. The companies were located in five different cities in Finland.

4.1.1 Software developers

All six interviewees acted in management roles in their companies. In addition, some additional job descriptions, such as design, analysis, product development and content provision, were mentioned by three interviewees. The job titles of the interviewees were technology manager, system manager, chief executive, account manager (who acted as a project manager), financial manager and development manager.

The interviewed software developers were located in three different cities. The home country of all companies was Finland. One of the companies was a part of an international concern and the head office of this company was located in the US. Half of the companies had one office in Finland and the other half had 2-6 offices. The number of branch offices abroad varied a lot since one company did not have a branch office abroad whereas two companies had several

dozens of branch offices. Two of the companies had one or two branch offices and one company had dozens of sales offices abroad.

One company was founded less than 5 years ago, two companies less than 10 years ago and the rest were more than 20 years old. There were three medium-sized companies with the total number of employees less than 50 and three companies with more than four hundred employees. The largest software developer had over 50 000 employees of which over 500 were in Finland. One company had its origin in the application area and software development was not the main part of its business. The net revenues of the companies varied from about 10 000 euros to over 100 million euros. Half of the companies had a net revenue less than two million euros whereas the other half had a net revenue more than 30 million euros.

The companies usually marked more than one application domain for their products and the application domains varied a lot. The most common application domains were business related software and system software / software tools which were mentioned by half of the companies. Embedded systems, knowledge based systems / expert systems and engineering and scientific software were mentioned by two companies and process control systems / process automation systems by one company. In addition, the application area of one company's products could be specified as entertainment.

The number of software projects was estimated either as the number of software projects per year or as the number of on-going software projects. There were less than five software projects in three of the companies, from 5 to 10 in one company and more than 10 projects in one company. In fact, one company had thousands of software projects at the concern level. One interviewee could not assess the number of the software projects.

Two companies had an ISO 9001 certificate at the concern level and one company did not use any kind of quality system. Three companies said that they used some kind of undocumented quality system or their quality system was under construction at the time. One company had had their maturity assessed but they did not have an official classification.

Four companies localised their products mainly in-house, i.e., they translated user interfaces and some of the documents into languages they spoke. However, all the companies outsourced some of the translation work to localisation vendors, for example, if the language was unknown to them or in case of some documents. Two companies said that their vision for the future was to outsource the whole localisation process.

4.1.2 Localisation vendors

All five interviewees from the four localisation companies acted in management roles. In addition, three of the five interviewees worked as a translator as well. One interviewee named technical marketing as his area of responsibility. Two interviewees were project managers. The rest of the job titles were software manager, localisation manager and chief technical officer.

The interviewed localisation vendors were located in two cities. The home country of the companies was Finland, except for one company whose home country at the concern level was the US. Two companies were a part of an international concern or belonged to a group of localisation companies.

Three of the four companies had one office in Finland and one company had two offices. One company had one branch office abroad and another operated only in Finland. Two companies had about 10 and 25 branch offices abroad respectively. One company was founded less than three years ago, the others over ten years ago. One company had a net revenue less than 1 million euros, another company less than 2 million euros and one company more than 2 million euros. One interviewee did not assess their net revenue.

The total number of employees in the localisation companies ranged from about ten to six thousand. However, the range in Finland was narrower since half of the companies had less than 25 employees and another half had no more than one hundred employees. The number of freelance translators used by the localisation vendors was not easy to estimate for all of the companies. The use of freelancers could be described by saying that half of the companies used freelancers' services only a little and the other half used them a lot.

Two companies estimated that the ratio between translation and localisation projects was 80-20% and one company assessed that the ratio was 50-50%. In addition, one company gained all of their net revenue from localisation projects, i.e., they did not do any translation if it was not a part of a localisation project. This company had included internationalisation in its services. This distinction can also be seen in the number of localisation personnel. The last-mentioned company defined all of their employees as localisation personnel whereas the other companies estimated that the portion of localisation personnel was between 20% and 40% of the total personnel.

The most common services of the localisation vendors were translation, desktop publishing, software localisation, web localisation and testing / quality assurance. These options were marked by all of the companies. Three companies listed also proofreading / reviewing as one of their services and two companies marked documentation and technical writing. All of the companies instructed or consulted their clients and/or students but they did not use terms education and consultation as such.

All the companies were specialised in translation concerning software and information technology in general as well as engineering. Commerce and finance were marked by three companies. All the other options were chosen by two companies except literature which was not chosen by anyone. Three companies considered their organisational approach to localisation as centralised and one company as decentralised.

The localisation vendors supported various numbers of languages and their working methods varied as well: one company translated only from foreign languages into Finnish in Finland whereas another company supported about 40 languages. The most typical supported languages, beside Finnish, were English and German.

4.2   Personnel and education

Next, we inquired the interviewees about their views on education and the skills of their personnel. In addition, we asked them if they were familiar with some books related to G11N and what kind of opinion they had of those books. Software developers were also encouraged to tell their experiences about localisation vendors and vice versa.

4.2.1 Software developers

All six software developers had employees specialised in project management and database design. Five companies had specialised roles for user interface designers and testers and four of the interviewees stated that they had people who specialised in system analysis and even in I18N issues.

The level of education varied but in most cases the personnel had technical education. University degrees such as Master of Science (M.Sc.) and Bachelor of Science (B.Sc.) in computer science or other technical discipline were quite typical degrees in two companies. In addition, half of the companies stated that they had many undergraduates in permanent positions. The employees of two companies had mainly other third level degrees.

Five companies gave in-house training for their employees while one company invested in self-education. The amount of in-house training varied from two to ten days per year. One company had decreed 40 hours to be the minimum amount for in-house training per year. The in-house training sessions usually dealt with technical issues such as new technologies or user training for new applications. Project management and improvement of activities were included in in-house training by two companies.

The interviewees were also asked to grade the foreign language skills of their personnel. Half of the interviewees said that their people spoke fluent and accomplished English and the other half ranked the English skills of their personnel as average or fair. One interviewee used a term "Engineer English" when describing the language skills of software developers. The other languages were used less frequently and these languages were not spoken as well as English by the employees. One company emphasised that they had excellent foreign language skills because of their multicultural human resources. Another company had non-stop language courses for their personnel.

There were quite different views of the future since half of the companies estimated that no new personnel would be hired within next three years and the other half expected the number of personnel to increase in some scale. Only one company believed in heavy expansion in the near future. New employees were needed mainly for programming and project management.

Software developers were asked to determine the most important factors that had affected their choice of a localisation vendor. They were also asked to describe briefly their experiences about localisation vendors. The most common factors in choosing a localisation vendor were knowledge of the target area, previous experiences and references. The second most common factors were price and reliability but the number of supported languages, familiarity and business relations were also mentioned as important factors. Software developers had mainly positive or fair experiences about localisation vendors. The problems encountered while working with localisation vendors will be described in section 4.4.1.

At the end of each interview, the interviewees were shown a list of books related to I18N and they were asked to comment the books if they had read them. The list of the books can be found at the end of Appendix 1. The interviewees were not too familiar with the books since only one of them recognised a few of the books: he had read two books and knew the title of one book.

4.2.2 Localisation vendors

The most common specialised role for a localisation team member was a project manager along with a translator. The second most common assignments were localisation engineering and testing. Recognition of cultural issues was considered to be a part of translators' work, not a specialised role, by half of the companies. In general, the interviewees underlined that employees should have knowledge of different kinds of assignments instead of one specific role. The importance of typing skills for productivity was emphasised by a couple of the interviewees.

The employees had usually a Master of Arts (M.A.) in translation in three of the four companies and a M.Sc. or B.Sc. in technical disciplines in one company. Two companies gave in-house training for their employees between two days and a week per year. One company estimated the amount of in-house training was as much as one and a half month per year. One company did not give in-house training as such but gave individual guidance instead when needed. In-house training sessions usually dealt with technical issues but office work and negotiation skills were also taught in some cases.

In contrast to the software developers, the localisation vendors were asked to grade the computing skills of their personnel instead of language skills. Three companies ranked the computing skills of their personnel as average or adequate for work and one interviewee described the computing skills of their personnel as excellent. However, there was a lot of variation in computing skills of the personnel. A general comment was that most of the personnel had average computing skills and they were also willing to learn more but there were still many employees with inadequate computing skills. Freelancers were credited with better than average computing skills by one interviewee.

Three localisation vendors did not expect the number of the personnel to increase within next three years. One company was looking forward to a considerable growth in the number of the personnel.

The interviewees were shown a list of books and asked to share their opinions of them. In this case the list consisted of books related to L10N, and it can be found at the end of Appendix 2. The results were similar to software developers since only one interviewee had read one of the books and two of them recognised three books by name.

4.3 Projects

The next set of questions concerned the projects in which the companies had been involved. Some technical questions were also asked in order to clarify environments the companies worked in. The questions were either open questions or multiple choice questions.

4.3.1 Development projects

The sizes of the development projects were estimated as calendar months and efforts in person months (PM). All the projects lasted from 6 months to 18 months. Half of the six companies assessed that the duration for a typical project was between 6 and 9 months. Another half of the companies estimated the duration from one year to one and a half years. Long projects were

often broken down into smaller subprojects. Typical estimate in efforts in PM was in the range from 9 to 120 PM. Typical efforts less than 20 PM were estimated by two companies, between 40 and 60 PM by three companies and about 120 PM by one company.

The sizes of the software products were estimated as lines of code (LOC). Typical number of LOC for two companies was less than 50 000 and for another two companies less than 100 000. One company had products with millions of LOC. One interviewee did not assess the number of LOC.

Native language of the products was in every case either English or Finnish. Half of the companies used always English while one company used always Finnish as a native language of the products. Two companies chose the language (Finnish or English) depending on the customer. The three most common languages the products were translated into were English, Finnish and German.

The main market areas of the products were the Nordic countries, Central Europe and Great Britain followed by the US and Asia. Two companies defined their markets as global. The most common reasons for localisation were meeting customers' needs and improving sales. Legislation in the target countries was mentioned as one reason by one interviewee.

Next, the interviewees were asked to describe the workflow of a typical development process concerning globalisation. Localisation needs seemed to be identified at some level during the analysis and design phases (e.g., in the design of database and in the elicitation of requirements) by most software developers. Only one interviewee said that they designed their products mainly for one country and made changes needed for localisation later. The rest of development processes did not differ much from the usual software development process. If the translation work was outsourced, only the character strings were sent to a localisation vendor.

The rest of the questions concerning development projects were multiple choice questions. The options for each question were "never", "occasionally" and "often" along with "not applicable / do not know / no comment" -option. The last-mentioned option was meant for the situations where the interviewee did not know the answer or the question could not be answered for some other reasons.

Most development projects started with customer requests. Five out of the six companies chose option "often" or "occasionally" for this question whereas technology facilities was the only starting point for one company. Not all interviewees could specify which life-cycle model they used, if any. The most common models were prototyping, waterfall and evolutionary.

The companies implemented their products for different platforms and environments using various technologies, programming languages and application generators as illustrated in Figure 1 and Figure 2. Windows and Unix / Linux were the most common platforms for the applications but platform independent products were also implemented often by two companies and occasionally by two other companies. One interviewee defined mobile applications as their typical product. This can be seen in Figure 1 in which "other" covers the mobile applications.
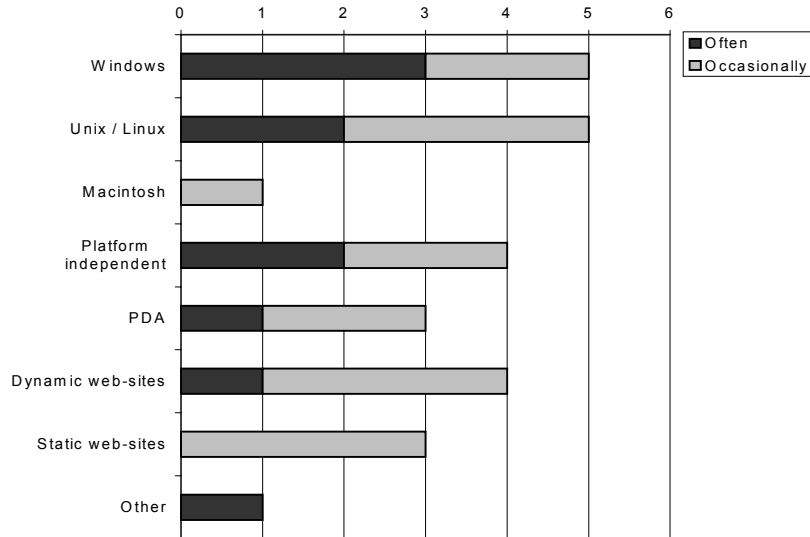
Figure 1. Platforms for applications in development projects.

Languages and technologies used in the implementation phase are shown in Figure 2. C / C++ and Java were the most common languages along with Visual Basic and HTML. Visual Basic was mainly used to create user interfaces of the products, and HTML was used for documentation purposes or web-site creation. However, web-sites can be seen as by-products because none of the companies focused merely on them. "Other" includes Symbian, Power Builder and PL/M among others.
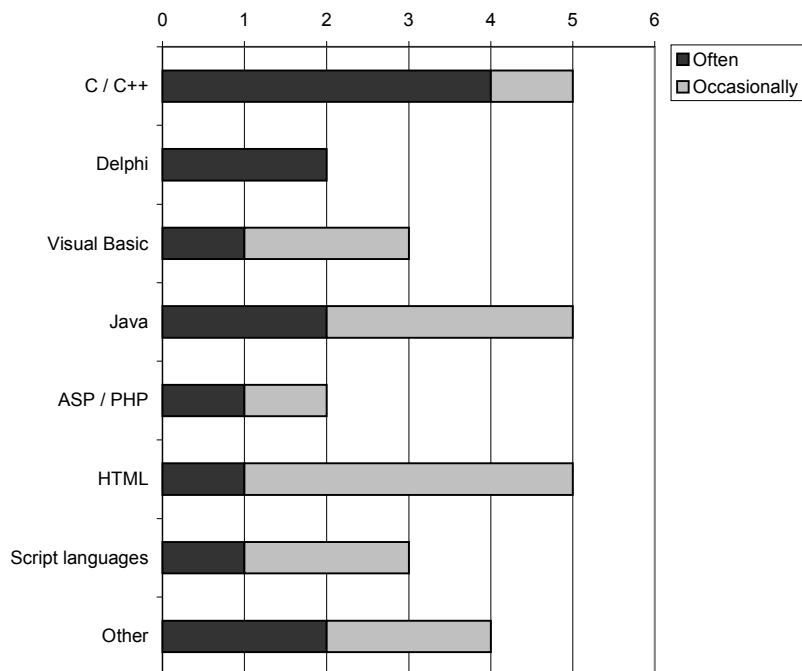
Figure 2. Languages and technologies in development projects.

The products included not only the application itself but also database, sample files and documents. Frequency of these components is presented in Figure 3. For information on localisation of the components, refer to Figure 6.
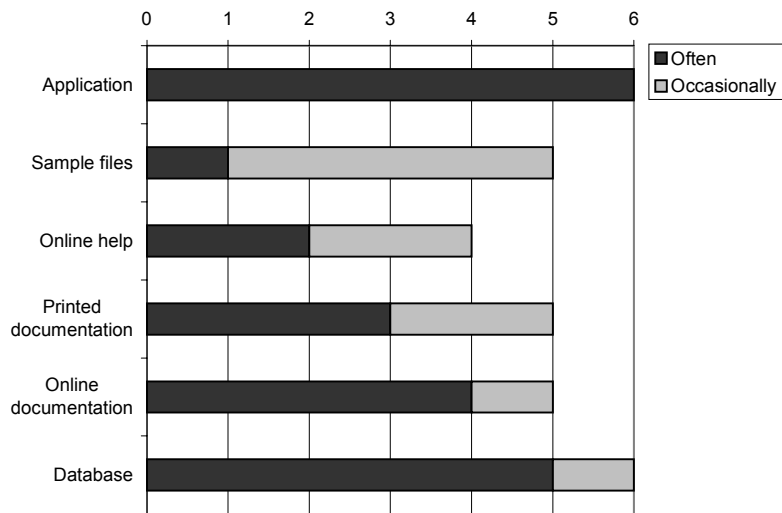
Figure 3. Components of products in development projects.

Figure 4 shows tools that were used during the development process. Programming environments and compilers were used often by all the companies. The second most common tools were test support tools and project management tools. Half of the companies often used some kind of version control to manage different versions of files during the development process. Image processing tools and desktop publishing tools were used only occasionally.
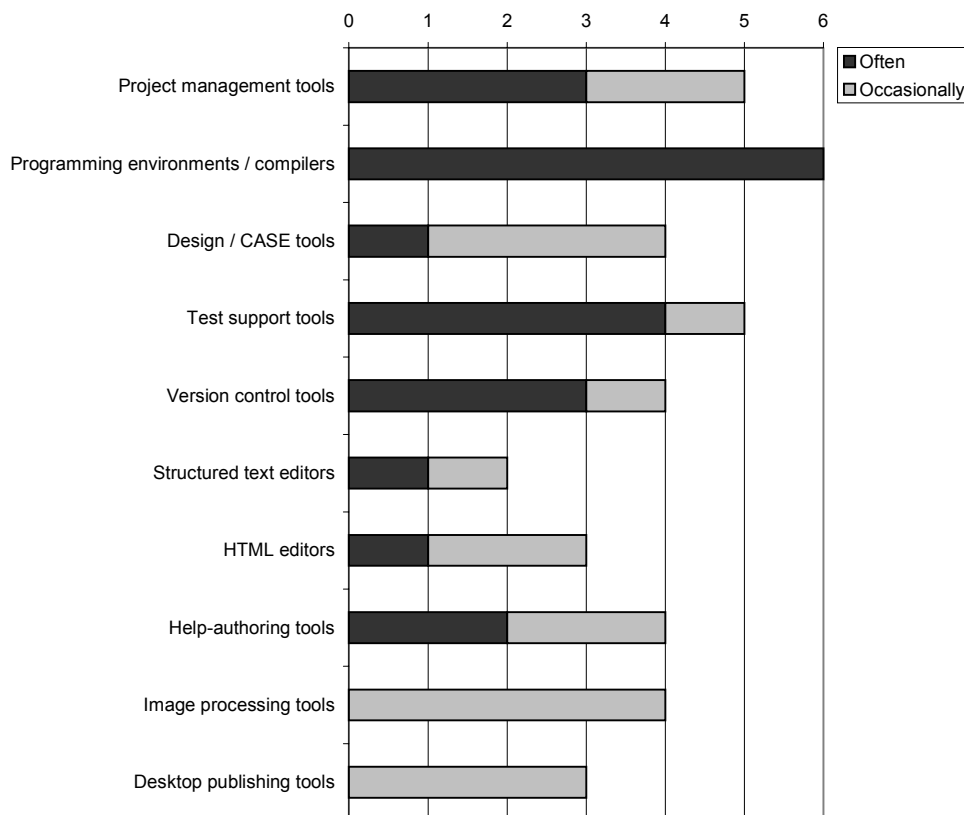
Figure 4. Tools in development projects.

Nearly all companies reported that they often or occasionally used usability engineering methods and techniques (e.g., user profiles, contextual task analysis and conceptual models) at some level during the development process. Despite of these user-centered design methods, only two companies often identified culture specific items during the development.

Four of the six software developers described that internationalisation was integrated into their development process, i.e., I18N issues were catered for during the analysis, design and implementation phases. On the other hand, internationalisation was considered as a separate process by two interviewees but they both stated that some of the I18N issues had been taken care of during the development.

The interviewees were asked some questions about internationalisation and user interface (UI) issues in order to clarify the level of globalisation. Only one company did not isolate language and culture specific items, such as strings and images, from the source code. The rest of the companies did it often or occasionally. The different techniques used to store language and culture specific items are presented in Figure 5. Different character sets and locales were supported often by two companies and occasionally by other two companies. Furthermore, half of the companies supported international hardware, such as keyboard. Two companies often avoided placing text in images as well as dynamic text messages, i.e., messages created by concatenating two or more character strings. In addition, a few interviewees answered "occasionally" to these questions.
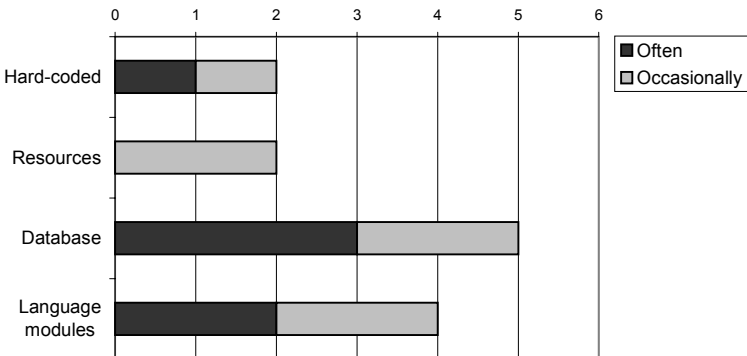
Figure 5. Storage of language and culture specific items in development projects.

Standard UI glossaries were rarely used since none of the companies used them often and only two companies used them occasionally. Two companies created their own product specific glossaries often and another two did this occasionally. The rest of the companies did not create glossaries at all. The goal of four companies was to design always an internationalised or modifiable user interface. Other two companies aimed for this kind of UI occasionally. One interviewee stated that they often customised culture specific UI elements to meet the needs of the target country. Two interviewees mentioned that they did it rarely. One company took the features of bi-directional languages into consideration in the user interfaces. Five companies recognised that text can expand during translation and allowed text expansion in the UI elements. One interviewee said that they did this occasionally.

Next, the interviewees were asked to consider the phases in which they made decisions about localisation and contacted a localisation vendor for the first time. Half of the companies said that they decided whether to localise the product before they started the development. Such a decision was made during the analysis or design phases by one company. Two companies did

not always make decisions about localisation until the product was finished, i.e., products were localised if there was a demand for it. In most of the cases localisation vendors were contacted after the product had been implemented which meant that no active communication or exchange of ideas between the software developers and localisation vendors existed before the completion of the products.

Software developers did not seem to consider localisation as a part of the development process since only one company often included L10N tasks in project schedules and L10N costs in budgets. In addition, one company included L10N tasks in project schedules and two companies included L10N costs in budgets occasionally. Most companies considered L10N as a separate process independently of the way it was made (in-house or outsourced). Requirements for localisation (e.g., quality assurance methods and technical requirements) were put down on paper by one company only. Half of the companies created some kind of instructions and guidelines for localisation vendors often and one company created such documents occasionally. Progress of localisation was monitored constantly by one company and occasionally by another company. Procedures for controlling change management were often created by one company while three other companies answered "occasionally" to this question.

Material was sent to a localisation vendor through the Internet in most cases. E-mail was used often by five companies and file transfer techniques, such as ftp, were used often by two companies. A paper copy was also delivered to a localisation vendor in some cases. Diskettes, CD-ROMs and zip disks were used quite rarely since only one company used them often and another one occasionally.

Figure 6 shows the components and Figure 7 the documents which were localised and translated during localisation process. Nearly all companies localised the applications, i.e., user interfaces, always or often while one company localised them only occasionally. Furthermore, help-files were localised more often than documentation or databases. If documentation existed, printed and online documentation was both translated equally. The most common localised documents were marketing materials and operating manuals followed by installation guides and getting started guides.
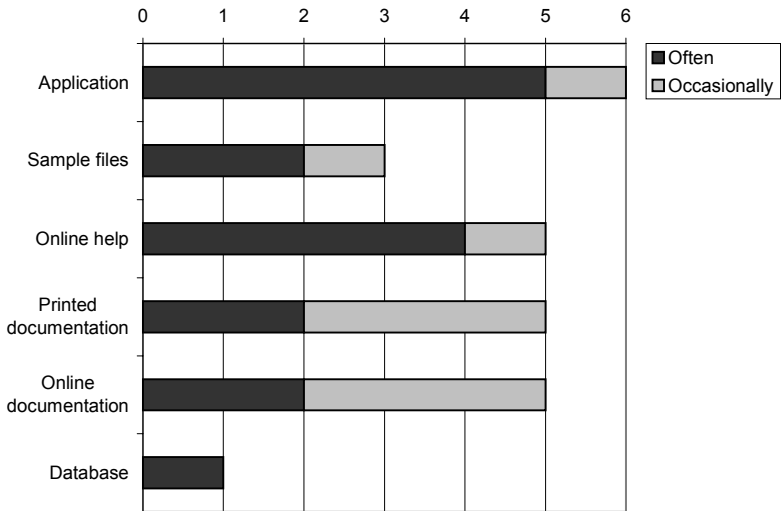
Figure 6. Localised components in development projects.
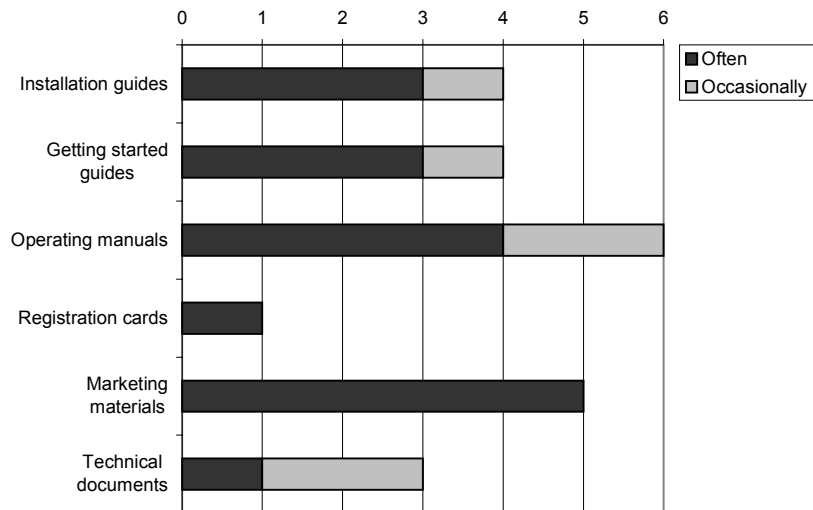
Figure 7. Translated documents in development projects.

The interviewees were asked to define the document formats they used in the companies in order to find out the possible compatibility or other type of problems related to the translation of documents. Word processor formats (e.g., DOC and WP), RTF and PDF were the most common formats but some other formats were also used as shown in Figure 8.
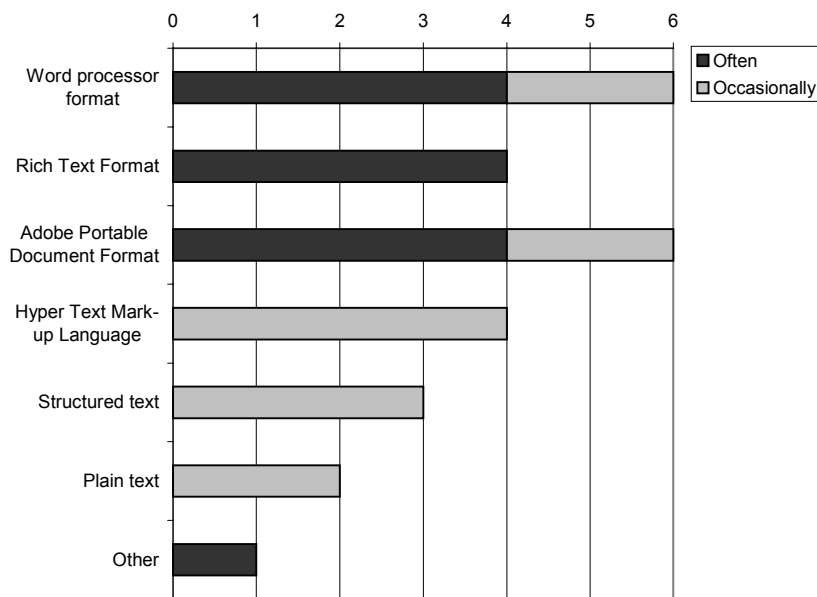
Figure 8. Document formats in development projects.

All companies used testing checklists as a part of their testing methods. Testing scripts were used often by one company and occasionally by three companies. Regression tests were scheduled often by one company and occasionally by two companies to ascertain that localisation had not damaged any features of products. In addition, release tests were performed often by two companies and occasionally by three companies. Four companies reviewed translated documents often and the rest of the companies did it occasionally.

Three companies released the original product before the localised variants. Another way of releasing products was used by two companies, as they preferred to release all variants at the same time. One interviewee told that they released all language variants at the same time with the exception of the Japanese variant that was released after the other variants. One company used both ways of releasing products.

Half of the companies were often and two companies occasionally satisfied with the localised products. The distribution of the answers was exactly the same for the question concerning the customers' satisfaction of the localised products. Four of the six companies were often or occasionally satisfied with the price-quality ratio of localisation. Only one interviewee stated that localisation was always too expensive. One interviewee did not share his opinion.

4.3.2 Localisation projects

The sizes of the projects were estimated as calendar months and effort in person months. Typical localisation projects lasted from a couple of weeks to three months. Updates were not considered as independent projects. Three of the four companies assessed that the duration for a typical project was between two and three months and one company estimated that it was from two weeks to one month. Typical efforts in PM were not easy to estimate because the number of participants and their work contributions fluctuated during the projects. Typical efforts can be roughly categorised in the following way: for two companies typical effort in PM was less than 10 PM, for one company less than 20 PM and for one company more than 20 PM.

The amount of translation work in the localisation projects was estimated as a number of translated words. Half of the companies assessed that a typical localisation project included from 10 000 to 20 000 words to be translated. One company estimated that there were more than 100 000 words to be translated in each project. One interviewee could not estimate the number of translated words but he underlined that it varied a lot depending on the product.

All the interviewees said that the most common native language of localised products was English. In addition, half of the companies localised quite often products that had been originally written in German. Three companies translated products mainly into Finnish and one company always into Chinese.

The workflow of a typical localisation project is shown in Figure 9. The figure was created on the basis of the interviews. Every project starts with the evaluation phase in which the received material is analysed and checked. The aim of this phase is to determine the size of the project using, for example, wordcounts, and identify black spots that need special knowledge or skills to be taken care of. The next phase is the project setup in which the project manager recruites team members, finds other resources and makes schedules. In this phase, the project terminology is usually created and required technical adjustments are done. After the project setup, translators start working on the material to be translated. The next phase, quality assurance, includes all methods (e.g., testing and proofreading) that are used to ensure that translations are consistent, appropriate and meet the customer's needs. Detected errors and bugs are then fixed before the delivery of the product.
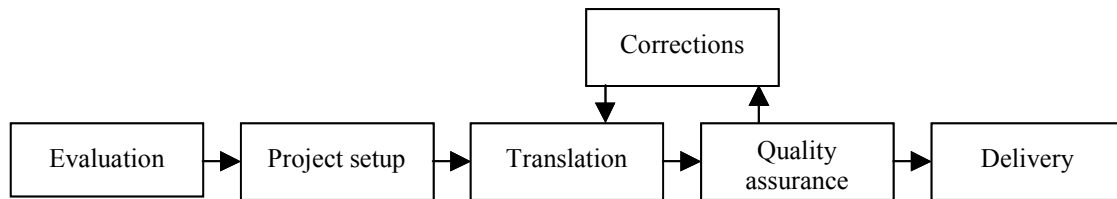
Figure 9. Workflow of a typical localisation project.

The rest of the questions concerning localisation projects were multiple choice questions. The options for each question were the same as for the software developers.

A Windows application was a typical localised product since all companies localised them often. Localisation of Macintosh applications was the main area of business for one company. Web-sites and web-based applications were also quite common products: one company localised them often and two companies occasionally. Device drivers, mobile applications and interactive TV-applications were mentioned as typical localised products by three companies respectively. Types of localised applications are shown in Figure 10.

Source code localisation and translation of language modules were the most common types of software localisation. Binary localisation and localisation of text-only resource files were also done often by one company and occasionally by one and two companies respectively. In Figure 11 "other" includes spreadsheets and other similar kind of solutions.
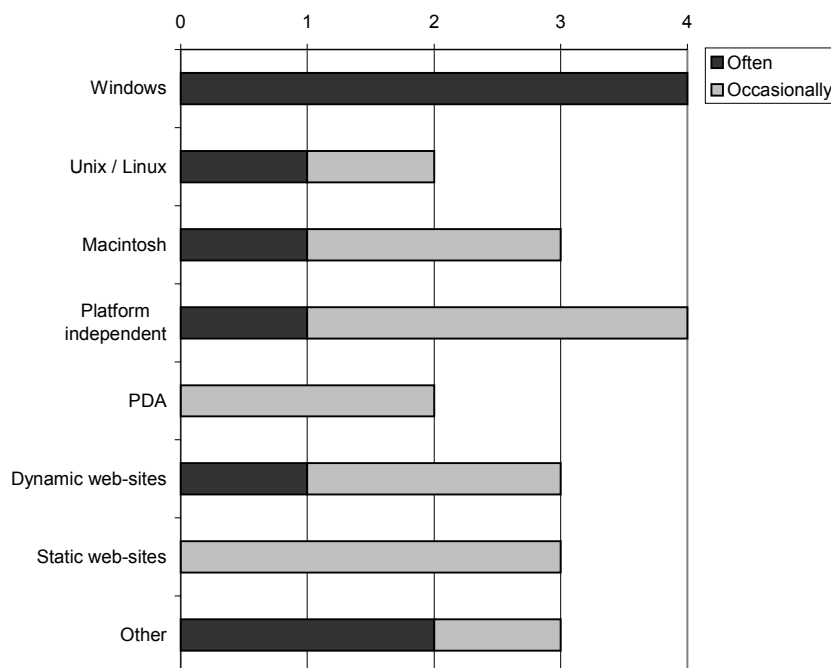
Figure 10. Platforms for localised applications in localisation projects.
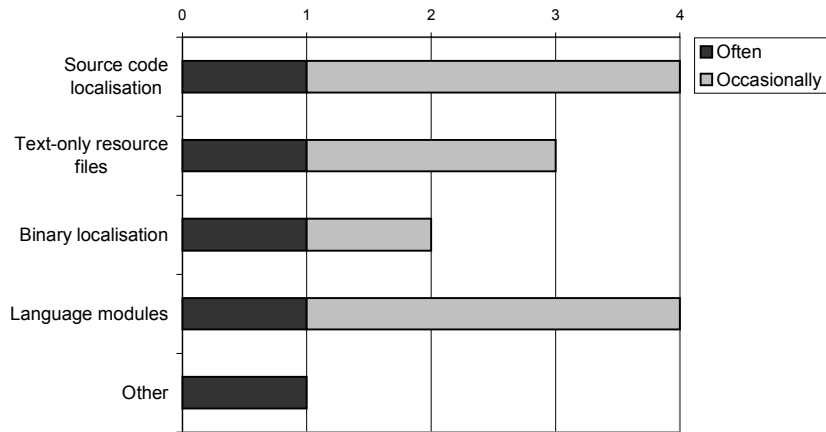
Figure 11. Types of software localisation in localisation projects.

Localised products contained several components as shown in Figure 12. Applications, online documentation and online help files were localised more often than the rest of the components. Databases were localised often only by one company and occasionally by another company.
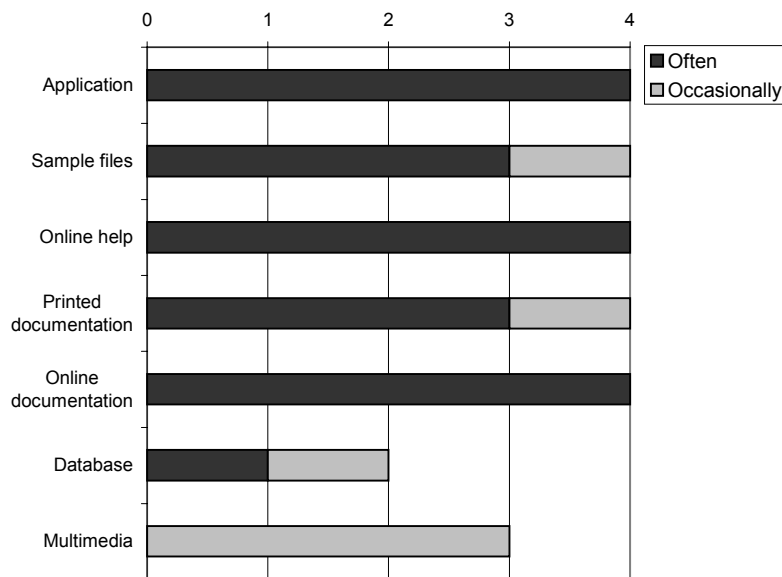
Figure 12. Localised components in localisation projects.

The four localisation vendors translated usually operating manuals, installation guides, marketing materials and getting started guides while registration cards and technical documents were translated less frequently (Figure 13). Educational material, licence agreements, warranty agreements and CD-covers were also mentioned as typical translated documents. The interviewees were also asked to describe in which format the documents were received and what kind of problems different formats had caused. As shown in Figure 14, a variety of formats were used but RTF seemed to be the most popular document format. "Other"-bar includes various formats of desktop publishing tools.
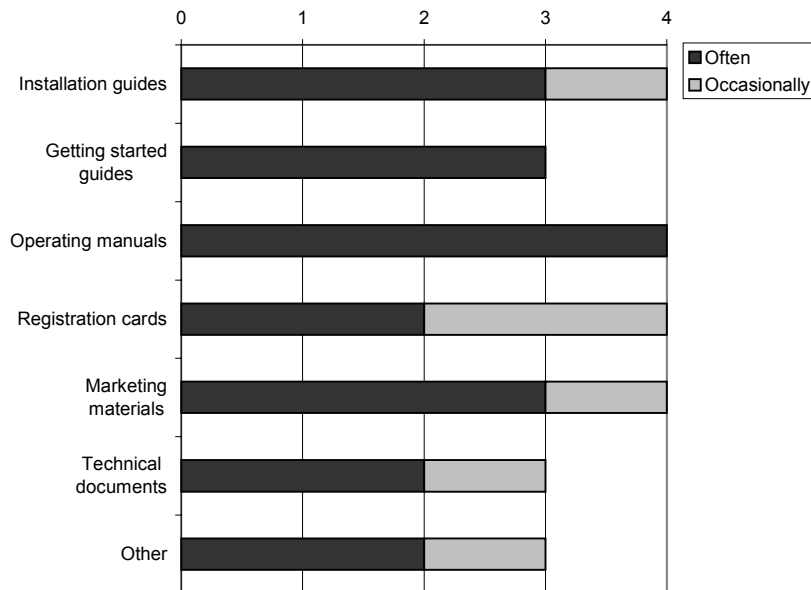
Figure 13. Translated documents in localisation projects.
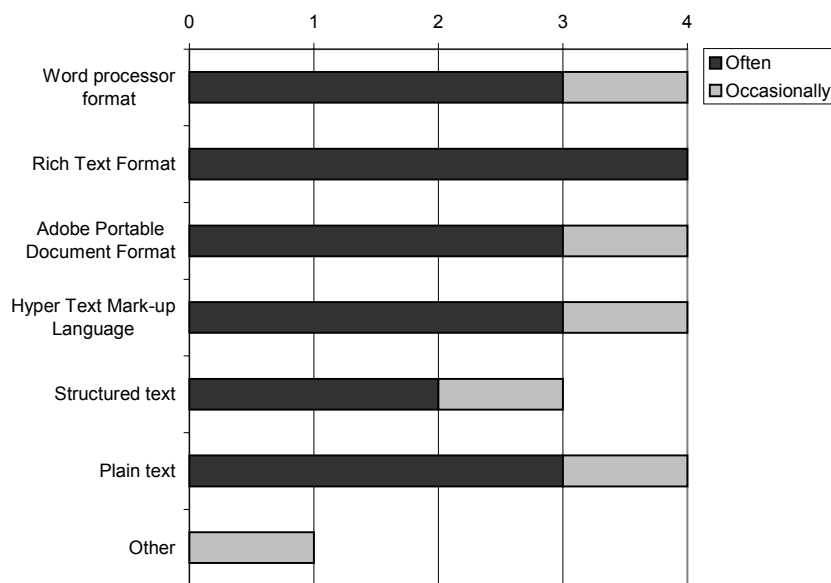
Figure 14. Document formats in localisation projects.

Next, the interviewees were asked some questions concerning project management. Accurate tasks and activities were defined often by half of the companies and occasionally by another half. Resources were allocated often by three companies and occasionally by one company. Three companies created project schedules often while one company did it occasionally. Moreover, half of the companies created project budgets often and another half occasionally. Two companies monitored the progress of localisation often (e.g., samples of translated text) and another two did this occasionally. Procedures for change management were used occasionally by two companies. Half of the companies documented localisation requirements often and another half occasionally. In addition, one company identified and documented risks often and three companies occasionally.

In most cases localisation vendors received material via the Internet by e-mail or file transfer techniques. Material was also received sometimes as a paper copy or on CD-disks by three companies.

Most localisation vendors used preparation checklists in every project to make sure that all important issues were taken into consideration. One company used such lists occasionally. All companies selected most appropriate tools for a project before it was started. Half of the companies created product specific glossaries often and another half occasionally. Test-compilation of source material was done often by one company only and occasionally by another company. Two companies used pseudo translation occasionally but the rest of the companies did not use such technique at all. Three companies created localisation kits or instructions for the translators often and one company occasionally.

Functionality and data of applications were seldom customised. None of the companies customised functionality often and only one company customised data often. However, customisation of functionality and data was done occasionally by half of the companies. Source code was edited often by one company and occasionally by another company.

Three companies took often advantage of previous translations by using translation memories (TM). One company did not use TM at all. Version control system was used often by one company and occasionally by two companies. None of the companies used machine translation (MT) tools. Tools used in localisation projects can be seen in Figure 15.

Three companies used always or often user interface glossaries and one company did not use them at all. Two companies modified user interfaces during localisation but only occasionally. Two companies had never modified user interfaces in order to achieve better usability for the users of the target culture. However, UI elements were resized often by two companies and occasionally by one company. The colours of user interfaces remained the same after localisation since only one interviewee said that they had occasionally changed the colours of the UI. Customisation of colours took place if the customer had asked the localisation vendor to adapt the product to the target market. Moreover, fonts were customised often by one company because they localised products to the Chinese market. One company customised fonts occasionally. Accelerators were customised often by two companies and not at all by the rest of the companies. Images and culture specific icons were edited quite rarely since only two companies stated that they had done it a few times.

Three companies used testing checklists often and one company occasionally. Testing scripts were less frequently used since only one company used scripts often and another one occasionally. Three companies informed software developers when they found problems or bugs by writing a problem report. All the companies proofread translated documents before delivery. Different kinds of tests performed by the localisation vendors are shown in Figure 16.
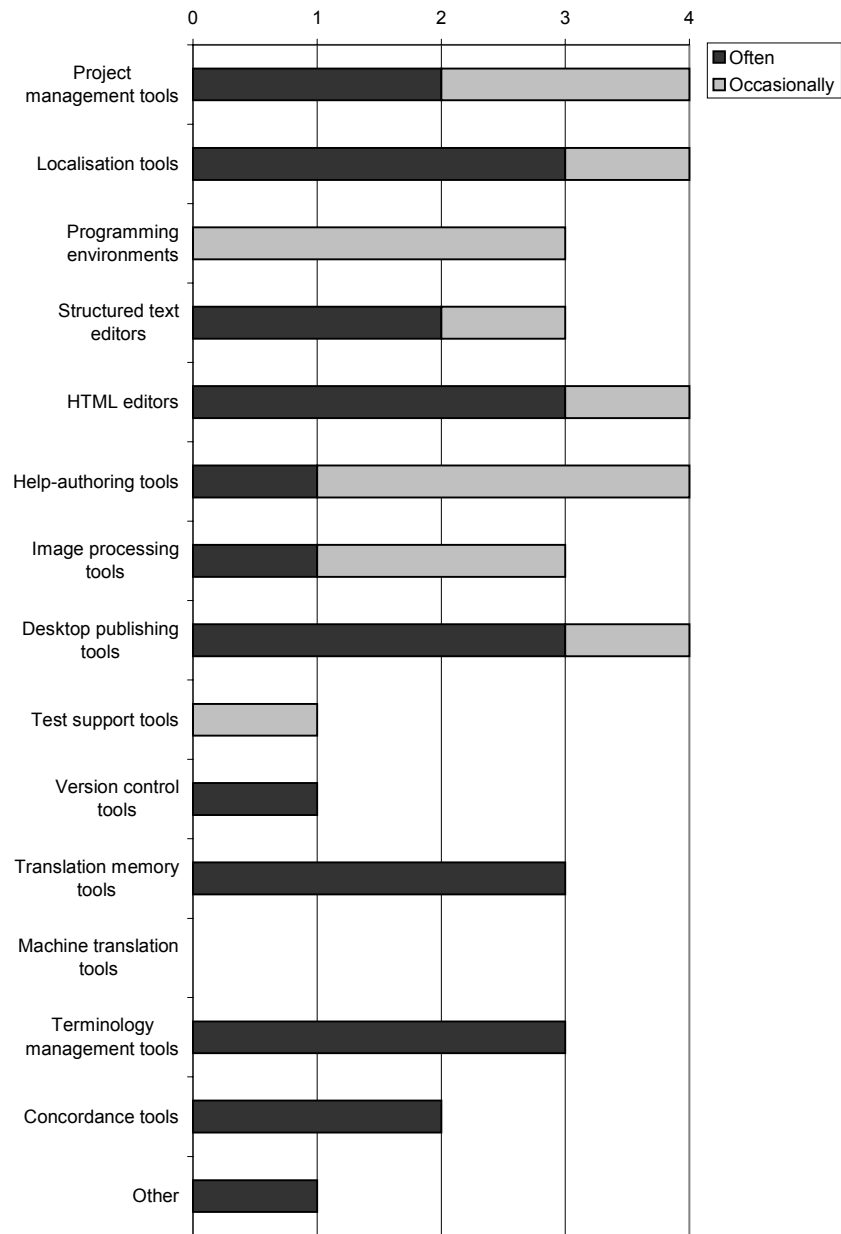
Figure 15. Tools used in localisation projects.
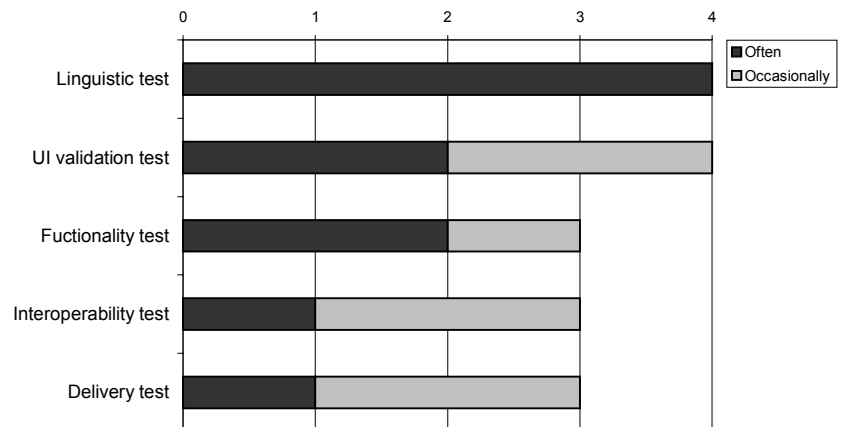
Figure 16. Testing of localised products in localisation projects.

The localisation projects finished often on-time and on-budget in half of the companies. Another half of the companies answered "occasionally" to this question. Three companies assumed that their customers were often satisfied with the localised products. In addition, two companies thought that their customers were often satisfied with the price-quality ratio of localisation. Another half of the companies answered "occasionally" to the question concerning the price-quality ratio of localisation.

4.4 Problems and expectations

Last in the project related discussions the interviewees were encouraged to describe problems they had experienced and to outline some suggestions for improving the current state of their practices. The interviewees were also asked to share their thoughts about the expectations they had of academic education and research in the area of software G11N.

4.4.1 Problems and development needs

Most problems of the software developers seemed to be related to project management. Some technical issues were also mentioned to cause problems whereas testing and communication, for example, were not considered as major problem areas. On the other hand, the localisation vendors felt that many major problems and client complaints were mainly caused by inadequate communication. The lack of suitable tools, strict schedules and frequent updates were also mentioned to inflict some problematic situations.

The software developers listed various problems related to project management. Sizes of projects were difficult to predict beforehand which led to changes in project budgets and schedules. For this reason, projects were behind schedules and translation work had to be done in haste. There were also problems in tracking development of projects which may have been partly due to the fact that the projects were not always documented until they were finished. According to one interviewee, the reason for the shortcomings of project management was that the company's ways of action were yet to be established.

The localisation vendors' opinions of project management varied. One interviewee said that their project management was not very systematic but it was not a major problem because projects were simple. Another interviewee stated that localisation had to be done hastily because software developers discovered the need for localisation too late and saw it only as an extra cost in projects. One localisation vendor was developing a project management tool of its own because, in their opinion, no suitable project management tool was currently available.

The physical distance between branch offices resulted in communication problems in case of two software companies and one localisation vendor. The distance between software developers and localisation vendors was also seen problematic by one localisation vendor because of indirect communication, i.e., communication between developers and translators was conducted by an intermediary. Moreover, three of the four localisation vendors pointed out that communication between software developers and localisation vendor was often quite problematic. According to them, software developers did not always provide adequate information for localisation or they

did not understand the overall localisation process and the prerequisites of a good translation. One localisation vendor acknowledged that they also had problems with internal communication.

Neither software developers nor localisation vendors considered human resources as a major problem area. One software developer interviewee admitted that the software engineers of the company had poor knowledge of G11N issues and he wished more education and experience for them in that area. Another interviewee stated that it was quite difficult to provide the overall picture of a software product to the members of the production team. Localisation vendors did not have major problems in recruiting personnel because there had been more applicants than there were open positions. However, one company had encountered difficulties in finding appropriate persons for their needs, and one interviewee felt that some of the personnel lacked the basic computing skills. One interviewee stated that some mistakes had been made in recruiting and human resource management without specifying it in detail.

Software developers listed some problems regarding the technical issues of internationalisation, such as internationalisation support of different programming languages and country specific operating environments. Software developers had also had problems with character sets and code set conversion. For example, one interviewee stated that they had had problems with the double byte character sets. However, at the time of the interviews only a few of the interviewees saw operating with different character sets as a problem because of the Unicode standard. Software developers also introduced some non-technical problems related to I18N, for example, lengthening of the translated text and choosing appropriate colours.

Testing was not considered as a major problem area. However, one company said that there had been minor problems with testing of all country specific features. In addition, one interviewee remarked that when testing was performed by people who could not understand the particular language it could decrease quality. This company localised products mostly in-house.

The biggest problem in the area of translation was the lack of context while localising a software product. According to the localisation vendors, many software developers provided only the user interface strings in text files or in spreadsheets to the localisation vendor, and usually no application, screen shots or other kind of contextual information was provided. For this reason, many localisation projects had been accomplished without any context which had had a significant impact on the quality of translation and resulted in lots of corrections. Also, the continuously changing terminology was considered problematic by two localisation vendors, and the difficulty in determining the terminology of a particular application domain was mentioned by one software developer. Two localisation vendors and one software developer regarded the length restrictions of character strings as a nuisance. Some software developers criticised localisation vendors for producing inaccurate translations of terms concerning the target area of the products.

Half of the localisation vendors regarded frequent updates as the most problematic issue in localisation. The problem was that in some cases the translation of a product was started before the product was finished which led to an update vortex. One localisation vendor suggested that hiring a consultant to organise the whole localisation process from the beginning would enhance the process considerably. However, he doubted the cost efficiency of such a solution. Inadequate internationalisation of software products was seen as a problem by three localisation vendors. One of them pointed out that searching and extracting hard-coded strings from the source code was not always an easy thing to do. One reason for this was the insufficient documentation of the source code.

The lack of good translation and localisation tools was considered as a big problem by most localisation vendors. The available tools were mentioned to be too expensive and some of the tools had proved to be inapplicable, i.e., they did not contain appropriate features for the purposes of the localisation vendors. In addition, not all of these tools were easy to use. One localisation vendor said that some software developers required localisation vendors to use special tools that were very expensive and could not be used for any other tasks. In contrast, one software developer criticised localisation vendors for not being able to handle different file formats. According to him, this had caused problems to software developers because they had to edit translations afterwards. Another software developer criticised localisation vendors for delivering the output in an incorrect format.

Finally, the interviewees were asked to specify the main problems when making a global software product. Elicitation of country specific requirements and the adaptation of the functionality of a product to different countries were mentioned to be difficult by one software developer. His solution was to specify and document the requirements of different areas as early as possible. Another software developer hoped to co-operate more closely with clients in order to deliver products that would satisfy customers' needs better. One software developer thought that finding an appropriate internationalisation technique was quite difficult. He stated that not all techniques were suitable for different kind of software projects and that some techniques may not be useful after a few years. Standardisation of I18N techniques was suggested as a solution to this problem. One interviewee said that programmers tended to forget that products were supposed to be global. In his opinion, education and experience would be effective cures for this problem. One interviewee saw the great number of localised products as a problematic issue.

One localisation vendor viewed the lack of standards regarding L10N as an essential problem for the whole localisation industry. Other main problems mentioned included insufficient instructions and working methods, frequent updates and maintenance of quality when work had to be done in haste and difficulties in evaluating workload. The most notable suggestions for improving the localisation process were developing a systematic and formal approach to the whole L10N process and starting a co-operation between software developers and localisation vendors earlier on the projects. Another improvement proposal was that localisation should be included in software developers' project plans. According to one interviewee, this would require more education on localisation matters for software developers.

4.4.2 Expectations of universities

More training in the area of software G11N was required for both software engineering students and translating students by all interviewees. Subjects for training suggested by software developers are shown in Figure 17 and by localisation vendors in Figure 18. The "other" bars were composed as follows. One software developer required more training in business economics for software engineering students and one localisation vendor wished that software engineers had better communication and project management skills. Importance of typing skills for translators was emphasised by two localisation vendors. A summary of these results can be seen in Figure 19.
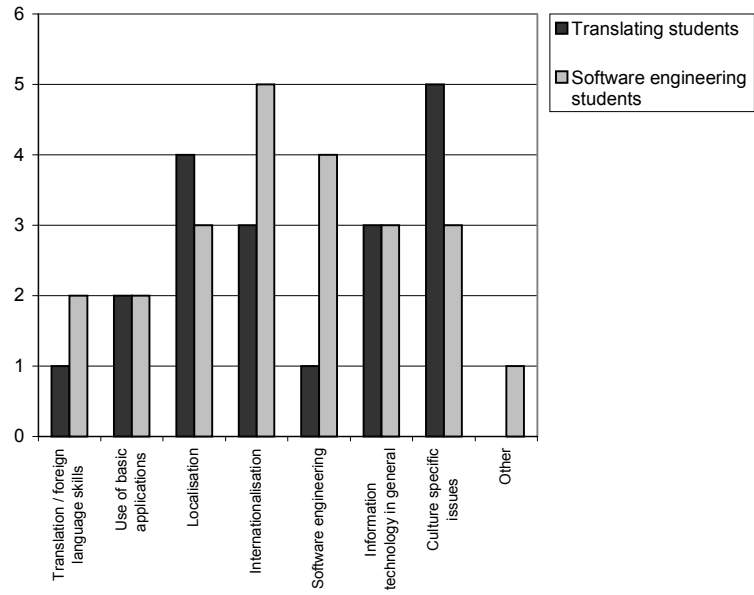
Figure 17. Training required by software developers.
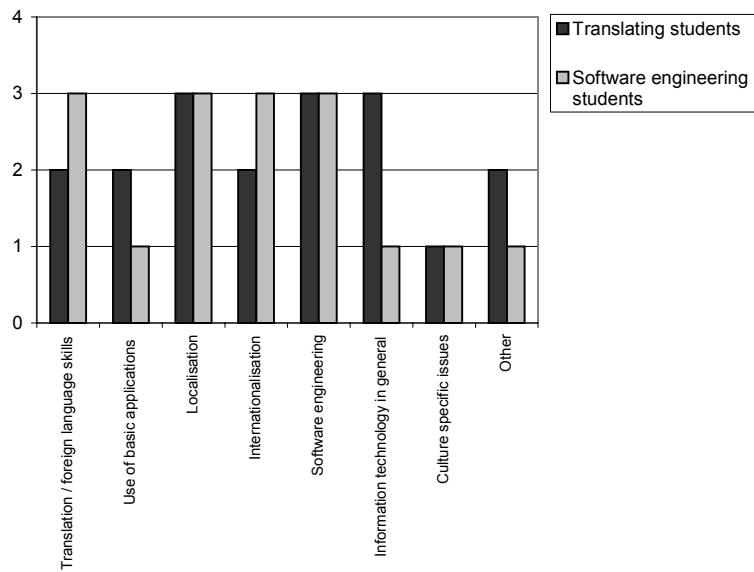
Figure 18. Training required by localisation vendors.
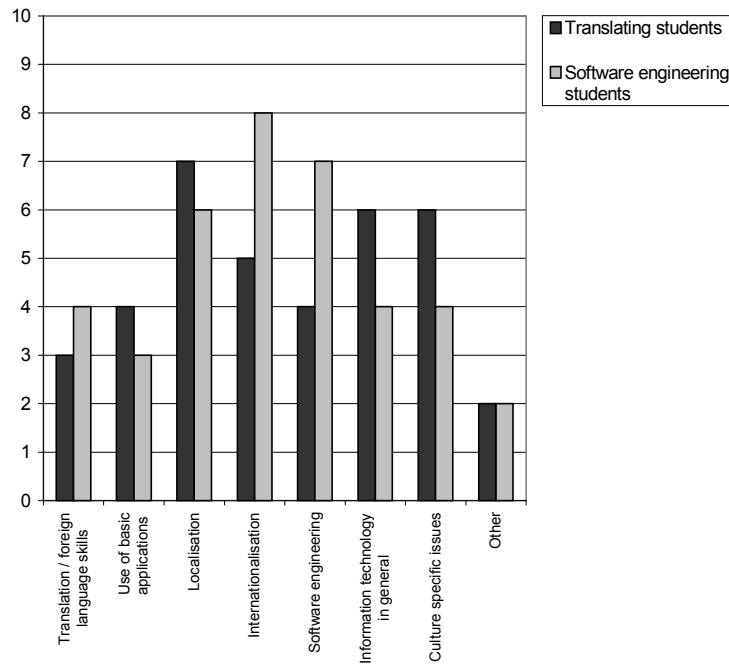
Figure 19. Summary of the required training.

All companies considered a Master's degree as a sufficient degree for translators but half of the companies also approved a Bachelor's degree. The distribution of the answers was nearly the same for the question concerning the level of education for globalisation personnel. A Master's degree was seen as an appropriate degree by all companies while a Bachelor's degree was also suitable for seven of the ten companies. One interviewee said that having good language skills was not enough for translators anymore. According to him, the best combination of education would be an academic degree with translation as a major and some technical discipline as a minor.

Seven companies expected academic research to develop new methods, techniques and models, and six companies hoped that universities should provide training for practitioners. Five companies considered consultation as one of the most important goals of academic research. All localisation vendors were willing to try production level tools but two of them were also interested in prototype tools. Half of the software developers did not want to try any tools developed in academic research, as they preferred to implement their own tools. Another half of the software developers considered both production level tools and prototypes to be appropriate for them.

The interviewees were also asked to propose new research topics regarding software G11N issues. Two software developers proposed research that would concentrate on the production of a global software product from scratch. Research on specific G11N issues, such as language support in terminal equipment, was also proposed during the interviews by a few software developers. In addition, there were some proposals that were more related to business matters than software production.

Localisation vendors' proposals included, for example, phases of a globalisation project and the ways localisation needs are catered for during software production. In addition, they hoped for an independent survey that would concentrate on comparing computer assisted translation tools

(e.g., translation memories) in order to find out the advantages and disadvantages of each tool. Two localisation vendors wished that academic research would provide new, compact and easy-to-use tools that should be designed to deal with particular tasks, such as pattern matching. One interviewee summarised the lack of good tools by saying "localisation tools are made to solve all the possible problems at the same time and for that reason they do not solve any problem properly".

5 DISCUSSION

This section includes the discussion of the survey findings. Section 5.1 describes components that are usually localised and discusses the sufficient localisation level. Section 5.2 sums up the current globalisation processes, and Section 5.3 concerns problems in producing a global software product. Section 5.4 concentrates on education needs and describes some proposals for research topics presented during the interviews.

There were some disparities in defining the terminology used in the survey, and in some cases the terminology was quite unknown to the interviewees. Considering this and the fact that only four of the eleven interviewees knew any of the books in the list containing some important volumes in this area, we can conclude that most practitioners have familiarised themselves with software globalisation by a learn-by-doing method, and thus their solutions for G11N problems are usually based on practice. Consequently, a lot of diversity exists in the operations of companies when producing a global software product.

5.1 Localised components

We start with considering which components of products are usually localised and how the practices used correspond to the issues 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 seems to be quite normal procedure and it corresponds to level 5 in Kano's (1995) localisation level classification that includes 7 different levels (Figure 20). Levels 6 and 7 were also reached in some projects.

Less risk,
less return

↓

More risk,
more return

1. Translate nothing
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
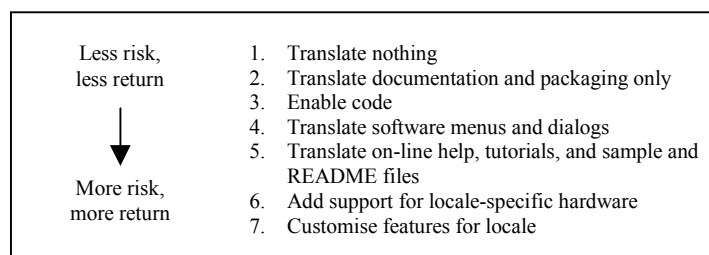
Figure 20. Levels of localisation (Kano, 1995).

Software developers tend to determine the localisation level by balancing risk and return (Kano, 1995). Translating components mentioned above 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 these cases, it is understandable to localise only the components that users are dealing with in order to reduce the costs and time-to-market. Furthermore, the main market-areas of products of the companies were Europe and the US which means easier globalisation than targeting, for example, to the Far East or the Middle East countries. According to Hietala & al. (2002), only 8 % of Finnish software developers had their first export country outside Europe and the US.

Usability, especially international usability, requires considering sufficient localisation level more carefully. If a software product is designed for world-wide delivery and for wide and heterogeneous audience, mere translation of the components may not always be enough but cultural adaptation is required, too. Cultural adaptation may seem to be a dispensable and time-consuming task and was often ignored or avoided by the interviewed companies. Consequently, the results of the survey support quite well del Galdo's and Nielsen's contention (1996) that most software developers take into consideration only the most obvious cultural differences, such as language and notations. Still, if one wants a software product to be truly adapted to a certain culture and to obey the models and ways of the target culture, one cannot ignore the deeper levels of culture.

Software developers approached international usability by designing an international 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. However, Ito and Nakakoji (1996) have argued that there is no universal interface that can be applied to any culture. According to Vine (2002), it is a known trade-off that making one product with the same user interface for the entire world means some sacrifices in usability. Even though nearly all of the software developers claimed to use usability engineering methods at some level during the development, problems in recognising cultural diversity still exist. This is quite understandable because most usability methods (see, e.g., Mayhew (1999)) do not take cultural diversity into consideration.

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 countries and cultures in question. In the third level, design must address specific cultural models that accommodate users' cultural characteristics. The practices described in the survey correspond to levels 1 and 2. Comparing the results of the survey with this classification and Kano's one, we can conclude that in many projects localisation equals to translation and no extensive cultural adaptation is usually done.

5.2 Globalisation process

We continue by concentrating on the software globalisation process and co-operation between software developers and localisation vendors. Software developers seem to make decisions about localisation quite early in the development life cycle. In fact, half of the interviewed software

developers made this decision before starting product development. Internationalisation is considered as a part of development process, rather than a separate process. This corresponds to views presented in the literature (e.g., O'Donnell, 1994; Hall, 2002) 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 a half of the software developers, 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 mentioned that they had just started producing global software products and some of them considered even the identification of G11N issues problematic.

Figure 21 shows the generalised G11N process from the development to release in the interviewed companies. The upper part of the figure consists of the phases conducted by software developers, and the lower part describes a simplified workflow of a typical localisation process (cf. Figure 9).
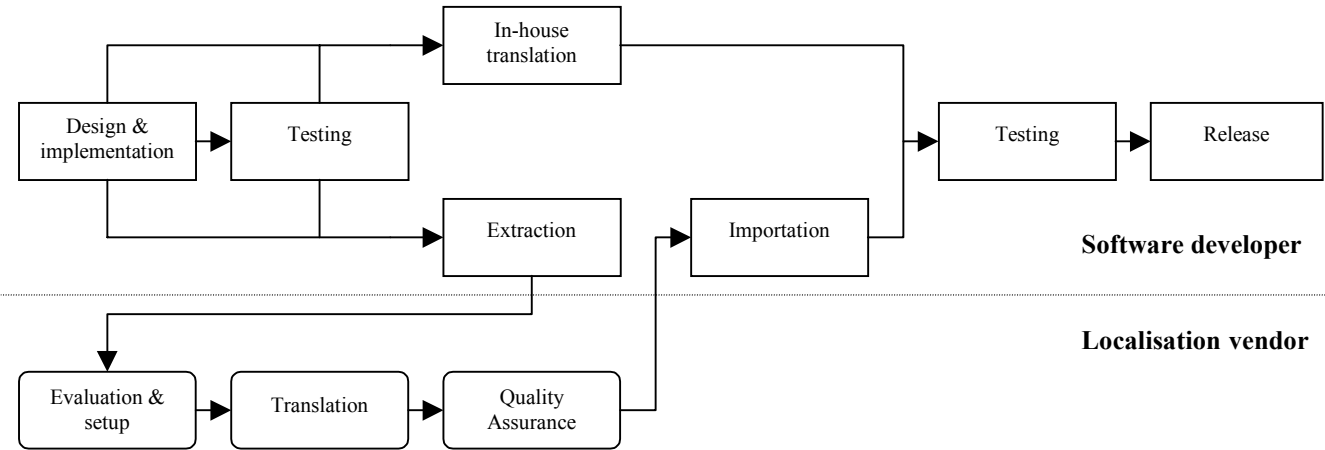
Figure 21. Typical globalisation process.

We found two approaches to localisation: in-house translation and outsourcing. Software developers do not view localisation as a part of the development process but rather as an extra cost. To reduce the costs, they localise products mainly in-house into the languages they speak and only some parts of localisation work are outsourced to localisation vendors. 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 software developers assessed the English language skills of their personnel as good or average, one localisation vendor thought that this practice could decrease the quality of translations. On the other hand, when accomplishing localisation in-house many problems concerning, for example, different tools and differences in file formats could be avoided.

There seems to be two basic reasons for outsourcing: either software developers do not speak the target language or they do not have time to localise all the components. Consequently, localisation seems to be considered purely as a developer's assignment and localisation vendors

are included into the process only if software developers can not localise all the components themselves. This approach may be one reason for not contacting localisation vendors until at the end of the development process (even if the decision about localisation have been made earlier) and for the fact that software developers usually do not include localisation in project schedules and budgets.

If the software developers outsource localisation, they usually deliver only the character strings to a localisation vendor. 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 localisation literature which usually focuses on binary or text-only resource-file localisation. In addition, this practice creates two additional steps to the G11N process: beside extracting strings, software developers need to import translated strings back into the product after translation. This practice is also problematic for translators because they can not benefit from the context information; an issue that the translators recognised as a significant problem resulting in lots of corrections. According to the localisation vendors, this problem stems from software developers' lack of understanding about translation.

Software developers' practices vary a lot but basic operations of localisation vendors do not differ as much. The working procedures as described in Figure 9 and Figure 21 follow quite well the localisation process described by Esselink (1998). Localisation vendors seem to have quite stable working methods, and the available technology is applied extensively. In fact, most localisation 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 localisation tools are used quite widely and glossaries, for example, are used to improve intelligibility of a product and ensure consistency with the environment and between different versions. However, software developers do not use glossaries when designing a product making glossaries less useful. Quality assurance methods vary from one project to another but all localisation vendors proofread all 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 localisation. If a localisation vendor receives only character strings, it is hard or impossible to do any testing apart from a linguistic test.

Communication and co-operation between software developers and localisation vendors seems to be slight and occasional despite of its importance as described by Rubric (2002), Esselink (1998) and Lingo Systems (2000). Communication is mainly restricted to business activities, and no extensive interaction between the participants exists. Software developers create a small localisation kit or simple instructions for localisation vendors, and localisation vendors report found bugs and detected internationalisation deficiencies to software developers. If some problems occur during localisation, localisation vendors contact software developers for help by using, for example, e-mail. More formal methods, such as documentation of localisation requirements and change management, are put into practice very rarely. This could partly stem from the simplicity of localisation projects but it may also reflect the immaturity of the processes. According to Topping (1999b), it is important that software engineering processes include internationalisation and localisation requirements when companies develop products for world-wide distribution.

5.3   Problems in globalisation

Next, we aggregate problems revealed by the survey into improvement needs. There seems to be problems related to G11N in every phase of product development, starting from analysis and continuing to testing, and affecting both software developers and localisation vendors. Improvement needs recognised in the survey can be roughly categorised as follows:

- Technical aspects of internationalisation
- Identification of globalisation issues
- Localisation practices
- Globalisation process

Technical aspects of I18N seem to be 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, some companies that have more experience in producing global software seem to consider the overall picture of G11N as a more problematic issue. For example, they had problems with identifying what kind of issues should be taken into consideration when producing a global software product and what kind of effects these issues 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.

Localisation practices and globalisation processes are partially intertwined since many localisation 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 software developers. Thus many problems in localisation 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. At the same time, a more systematic approach (e.g., project management, communication, co-operation) to globalisation could improve the current state of the practices.

5.4   Expectations of universities

The interviewees were asked to assess education needs for both software developers and localisation vendors. More training was generally required for students and practitioners in the area of globalisation and software engineering. Figure 22 presents a summary of education needs. The areas in which training is most needed are marked with "*" and less important areas with "**". A "***" indicates areas in which no additional training is needed.

	Software engineering students	Translating students
Translation / foreign language skills	**	***
Basics of information technology	***	**
Localisation	**	*
Internationalisation	*	**
Software engineering	*	**

Figure 22. Sufficiency of current education.

As expected, more training in internationalisation is needed for software engineering students. This is fairly obvious since no internationalisation courses are currently available and many problems related to I18N exist as described in the previous section. Also, more education in software engineering and localisation were commonly wished for software engineering students in the interviews. These needs were mainly emphasised by the localisation vendors, presumable at least partly due to indefinite G11N processes and the problems in co-operation between software developers and localisation vendors. Software engineering courses are being lectured in many universities for software engineering students but localisation courses are usually targeted for translating students.

Education needs for translating students were quite predictable, as localisation and information technology were the most wanted subjects of training. Nowadays having good language skills is not enough for translators but technical skills are also needed. More training was also required in culture specific issues for translating students, included in localisation education in Figure 22. Beside having good language skills, localisers should have good knowledge of the features of the target culture (Taylor, 1992; Lingo Systems, 2000). Surprisingly, only one localisation vendor but almost all software developers felt that there is a need for more training in culture specific issues for translating students. This is surprising because many localisation vendors felt that software developers did not want them to get involved in product development apart from translation. Familiarity with a target culture was considered as a part of translator's all-round education by the localisation vendors. Of course, software engineers should also have some knowledge about culture specific issues. This need is included in internationalisation education in Figure 22

As a summary, practitioners have some knowledge of globalisation but more education and experience is needed for both software developers and localisation vendors. All the companies saw a Master's degree or a Bachelor's degree to be appropriate for their employees, so it could be a good idea to include this kind of education in basic studies of university degrees.

In addition to education, the interviewees were asked to give proposals regarding new research topics related to software G11N and to specify which kind of new tools they would like to have. Those questions did not create much conversation because the questionnaire forms were quite extensive and the questions were asked at the end of each interview. However, the proposals mentioned by the interviewees reflect on the information needs for the companies somehow and some ideas for further research were also received.

The basic problem for many software developers seems to be the identification of the G11N issues that need to be taken into consideration when producing global software. Additionally,

applying different techniques to practice was not considered as an easy task. Therefore, it is understandable that one common proposal for research topic was the production of a global software product. This is quite extensive topic but many interesting subtopics can be found. Localisation vendors seem to have similar research interests since their proposals were almost the same as those proposed by the software developers but they approached the topics from a different viewpoint. Localisation vendors were also interested in localisation and translation tools. Their interest in those tools is not surprising as current localisation tools are expensive, complicated to use and do not include the proper features. Several tools have been presented and compared in the literature (e.g., Waßmer, 2002; Esselink, 1998) but often those presentations are quite short and limited in scope.

6   CONCLUSIONS

Importance of software globalisation is increasing as software companies continue expanding their market-areas, and more localisation services will be needed in the future. However, there is not much literature available about software globalisation processes and practices at the industry level. In this survey, we interviewed ten industry representatives to find out the current state of the software globalisation practices in the Finnish software industry. Although the sample was quite small, some interesting findings were obtained.

Globalisation practices vary from one company to another, and solutions seem to be based mainly on experience. Consequently, no consistent techniques or methods exist between the companies, even though most software developers have integrated internationalisation issues in the product development. In contrast, software developers do not view localisation as a part of the development process but rather as an extra cost. To reduce the costs, software developers localise the products mainly in-house into languages they speak.

In many projects localisation equals to translation and no extensive cultural adaptation is usually done. In a software developer's opinion, localisation vendors' only task is to translate the product and its components, and they do not expect localisation vendors to take care of any other aspects of adaptation. If some additional adaptation beside translation is needed, it will be handled by the software developer.

Globalisation related problems occur in every phase of software development. Many problems concerning localisation of software products seem to be caused by inadequate and occasional communication and co-operation between software developers and localisation vendors. Practitioners have knowledge of globalisation but more education and experience is needed for both software developers and localisation vendors.

ACKNOWLEDGEMENTS

REFERENCES

Aberdeen Group (2001): *Offshore Software Development - Localization, Globalization and Best Practices in an Evolving Industry.* Available on request at `http://www.aberdeen.com` (accessed 10.6.2002).

Apple Computer Inc. (1992): *Guide to Macintosh Software Localization*. Addison-Wesley Publishing.

Bradley R.K. (2001): Quelle est la clef à la mondialisation? *Proceedings of International Symposium on Technology and Society*, IEEE Computer Society, 117-124.

Deitsch A.& Czarnecki D. (2001): *Java Internationalization*. O'Reilly & Associates.

Del Galdo E.M. & Nielsen J. (eds.)(1996): *International User Interfaces*. John Wiley & Sons.

DePalma D., McCarthy J.C. & Armstrong A. (1998): *Strategies for global sites*. Forrester Research, **3**(3).

Esselink B. (1998): *A Practical Guide to Software Localization*. John Benjamins Publishing.

Farid M. (1990): Software engineering globalization and localization. *Proceedings of Systems, Man and Cybernetics*, IEEE Computer Society, 491-495.

Global360 (2002): *Taking Software Global: Successful Strategies for Product Development*. Available on request at `http://www.global360.com/research/tsg/Results.htm` (accessed 12.11.2002).

GlobalSight (2000): *Best Practices of Global eBusiness Leaders*. GlobalSight Corporation. Available on request at `http://www.globalsight.com` (accessed 10.6.2002).

Hall B. (2002): Developing Software with Internationalization in Mind. *Internationalization - The Annual Guide from Multilingual Computing & Technology*, April/May 2002, 8-10.

Harris J. & McCormack R. (2000): *Translation is not enough. Considerations for global Internet development.* Sapient Report.

Hietala J., Maula M., Autere J., Lassenius C. & Autio E. (2002): *Finnish Software Product Business: Results from the National Software Industry Survey 2002*. Helsinki University of Technology.

Hofstede G. (1997): *Cultures and organisations: software of the mind*. McGraw-Hill.

Hoft N. (1996): Developing a Cultural Model. *International User Interfaces,* edited by del Galdo E.M. & Nielsen J., John Wiley & Sons, 41-73.

Kano N. (1995): *Developing International Software for Windows 95 and Windows NT*. Microsoft Press.

Kaplan M.S. (2000): *Internationalization with Visual Basic*. SAMS Publishing.

Kokkotos S. & Spyropoulos C.D. (1997): An Architecture for Designing Internationalized Software. *Proceedings of the 8th International Workshop on Software Technology and Engineering Practice*, IEEE Computer Society, 13-21.

Lingo Systems (2000): *The Guide to Translation and Localization - Preparing Products for the Global Marketplace*. Available on request at `http://www.lingosys.com` (accessed 26.6.2002).

Localisation Industry Standards Association (2001): *The US Technology Globalization Survey. Globalization & Localization Services, Markets and Trends*. Available on request at `http://www.lisa.org/interact/2001/geomarket.html` (accessed 21.8.2002).

Marcus A. & Gould E.W. (2000): Crosscurrents: cultural dimensions and global Web user-interface design. *ACM Interactions*, **7**(4), 32-46.

Mayhew D.J. (1999): *The usability engineering lifecycle : a practitioner's handbook for user interface design*. Morgan Kaufmann Publishers.

Nielsen J. (1996): International Usability Engineering. *International User Interfaces,* edited by Del Galdo E.M. & Nielsen J., John Wiley & Sons, 1-19.

Nikula U., Sajaniemi J. & Kälviäinen H. (2000): *A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises*. Telecom Business Research Center Lappeenranta, Lappeenranta University of Technology.

O'Donnell S.M. (1994): *Programming for the World: A Guide to Internationalization*. Prentice Hall.

Piroumian V. (1997): Internationalization Support in Java. *IEEE Micro*, **17**(3), 20-29.

Plumley D.O. (2000): *Global eCommerce - The Market, Challanges & Oppurtinities*. Bowne Global Solutions.

Rafii F. & Perkins S. (1995): Internationalizing Software with Concurrent Engineering. *IEEE Software*, **12**(5), 39-46.

Rubric (2002): *A Guide to Localization Management*. Available at `http://www.lisa.org/2002/pdfs/Rubric_L10nGuide.pdf` (accessed 30.5.2002).

Russo P. & Boor S. (1993): How Fluent is Your Interface? Designing for International Users. *Proceedings of INTERCHI'93*, ACM Press, 342-347.

Sheridan E.F. (2001): Cross-Cultural Web Site Design. *MultiLingual Computing & Techonology*, **12**(7). Available at `http://www.multilingual.com` (accessed 29.1.2002).

SIL International (2002): *Ethnologue*. WWW-site, `http://www.ethnologue.com` (accessed 27.09.2002).

Taylor D. (1992): *Global Software. Developing Applications for the International Market*. Springer Verlag.

Topping S. (1999a): Organizing Localization In Large Companies. *MultiLingual Computing & Techonology*, **10**(6). Available at `http://www.multilingual.com` (accessed 29.1.2002).

Topping S. (1999b): Evaluating Your Corporate Localization Methods. *MultiLingual Computing & Techonology*, **10**(5). Available at `http://www.multilingual.com` (accessed 29.1.2002).

Uren E., Howard R. & Perinotti T. (1993): *Software Internationalization and Localization*. Van Nostrand Reinhold.

Vine A. (2002): Designing User Interfaces for an International Audience. *Internationalization - The Annual Guide from MultiLingual Computing & Technology*, April/May 2002, 15-22.

Waßmer T. (2002): Comparing Tools Used in Software Localisation. *MultiLingual Computing & Techonology*, **13**(6). Available at `http://www.multilingual.com` (accessed 9.10.2002).

Yeo A. (2001): Global-Software Development Lifecycle: An Exploratory Study. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 104-111.

Yunker J. (2002): *Beyond Borders - Web Globalization Strategies*. New Riders Publishing.

Zahed F., van Pelt V. & Song J. (2001): A Conceptual Framework for International Web Design. *IEEE Transactions on Professional Communication*, **44**(2), 83-103.

APPENDIX 1: Questionnaire form for software developers

Software Globalisation in Finland: A State-of-the-practice Survey
Questionnaire form for software developers

1 GENERAL INFORMATION

1.1   Place _____
1.2   Time _____
1.3   Participant_____
1.4   Occupation _____
1.5   Job description
         ❑   Management
         ❑   Design / programming
         ❑   Other, please specify _____

2 BACKGROUND INFORMATION OF THE COMPANY

2.1   Company name _____
2.2   Home country_____
2.3   Number of offices in Finland _____
2.4   Number of offices abroad _____
2.5   Year established _____
2.6   Net revenue _____
2.7   Number of software projects _____
2.8   Application domain
         ❑   Business software
         ❑   System software / software tools
         ❑   Process control system / process automation system
         ❑   Embedded system
         ❑   Engineering and scientific software
         ❑   Knowledge based system / expert system
         ❑   Other, please specify _____
2.9   Level of quality system _____
2.10   Organisational approach to localisation
         ❑   In-house
         ❑   Outsourced

3 PERSONNEL

3.1   Number of personnel (total and in Finland)_____
3.2   Development team structure, specialised people in
         ❑   Project management
         ❑   System analysis
         ❑   User interface design
         ❑   Database design
         ❑   Technical writing
         ❑   Testing
         ❑   Internationalisation issues
         ❑   No specialised job descriptions
         ❑   Other, please specify _____
3.3   Typical personnel education / degrees _____
3.4   In-house training per year _____
3.5   Subjects of training sessions _____
         _____
3.6   Language skills of the personnel _____
         _____
3.7   Need for new personnel within 3 years (number and fields of specialisation) _____
         _____

4 DEVELOPMENT PROJECTS

4.1   Size of projects in person weeks/months (typical and range) _____
4.2   Duration in weeks/months (typical and range) _____
4.3   Software size in lines of code (typical and range) _____
4.4   Market areas of products _____
4.5   Typical native language of products _____
4.6   Number of language variants _____
4.7   Languages products are translated into _____
         _____
4.8   Reasons for localisation of products _____
         _____
         _____

APPENDIX 1: Questionnaire form for software developers

4.9   Workflow of a typical development process concerning globalisation _____

_____

_____

1=not applicable / do not know / no comment, 2=never, 3=occasionally, 4=often

			1	2	3	4
4.10		Project starting point				
	4.10.1	Customer requests	❑	❑	❑	❑
	4.10.2	Technology facilities	❑	❑	❑	❑
	4.10.3	Other, please specify	❑	❑	❑	❑
4.11		Life-cycle model				
	4.11.1	Waterfall	❑	❑	❑	❑
	4.11.2	Evolutionary	❑	❑	❑	❑
	4.11.3	Spiral / incremental	❑	❑	❑	❑
	4.11.4	Prototyping	❑	❑	❑	❑
	4.11.5	Other, please specify	❑	❑	❑	❑
4.12		Components of software products				
	4.12.1	Application	❑	❑	❑	❑
	4.12.2	Sample files	❑	❑	❑	❑
	4.12.3	Online help	❑	❑	❑	❑
	4.12.4	Printed documentation	❑	❑	❑	❑
	4.12.5	Online documentation	❑	❑	❑	❑
	4.12.6	Database	❑	❑	❑	❑
	4.12.7	Other, please specify	❑	❑	❑	❑
4.13		Miscellaneous issues related to development				
	4.13.1	Use of usability engineering methods and techniques	❑	❑	❑	❑
	4.13.2	Identification of culture specific items	❑	❑	❑	❑
	4.13.3	Creation of product specific glossary	❑	❑	❑	❑
	4.13.4	Use of version control	❑	❑	❑	❑
	4.13.5	Other, please specify	❑	❑	❑	❑
4.14		Internationalisation				
	4.14.1	Internationalisation integrated into development	❑	❑	❑	❑
	4.14.2	Internationalisation as a separate process	❑	❑	❑	❑
	4.14.3	Isolation of language and culture specific items	❑	❑	❑	❑
	4.14.4	Supporting of different character sets	❑	❑	❑	❑
	4.14.5	Supporting of locales	❑	❑	❑	❑
	4.14.6	Supporting of international hardware	❑	❑	❑	❑
	4.14.7	Avoidance of dynamic text messages	❑	❑	❑	❑
	4.14.8	Avoidance of text in images	❑	❑	❑	❑
	4.14.9	Other, please specify	❑	❑	❑	❑
4.15		User interface tasks				
	4.15.1	Use of UI glossaries	❑	❑	❑	❑
	4.15.2	Design of modifiable/internationalised UI	❑	❑	❑	❑
	4.15.3	Design of multilingual UI	❑	❑	❑	❑
	4.15.4	Supporting of bi-directional languages in UI	❑	❑	❑	❑
	4.15.5	Allowing of text expansion in UI elements	❑	❑	❑	❑
	4.15.6	Customisation of culture specific UI elements	❑	❑	❑	❑
	4.15.7	Other, please specify	❑	❑	❑	❑
4.16		Phase in which the decision about localisation is made				
	4.16.1	Before starting product development	❑	❑	❑	❑
	4.16.2	Analysis / design	❑	❑	❑	❑
	4.16.3	Implementation	❑	❑	❑	❑
	4.16.4	After implementation	❑	❑	❑	❑
	4.16.5	Other, please specify	❑	❑	❑	❑
4.17		Phase in which the localisation vendor is contacted				
	4.17.1	Before starting product development	❑	❑	❑	❑
	4.17.2	Analysis / design	❑	❑	❑	❑
	4.17.3	Implementation	❑	❑	❑	❑
	4.17.4	After implementation	❑	❑	❑	❑
	4.17.5	Other, please specify	❑	❑	❑	❑
4.18		Release schedule				
	4.18.1	Original first, variants later	❑	❑	❑	❑
	4.18.2	All at the same time	❑	❑	❑	❑
	4.18.3	Other, please specify	❑	❑	❑	❑

APPENDIX 1: Questionnaire form for software developers

4.19   Localisation related issues
    4.19.1   Localisation included in project schedule   ❑ ❑ ❑ ❑
    4.19.2   Localisation costs included in project budget   ❑ ❑ ❑ ❑
    4.19.3   Documentation of localisation requirements   ❑ ❑ ❑ ❑
    4.19.4   Creation of instructions and guidelines for localisation vendor   ❑ ❑ ❑ ❑
    4.19.5   Monitoring of progress   ❑ ❑ ❑ ❑
    4.19.6   Creation of change management procedures   ❑ ❑ ❑ ❑
    4.19.7   Other, please specify   ❑ ❑ ❑ ❑

4.20   Localised components
    4.20.1   Application   ❑ ❑ ❑ ❑
    4.20.2   Sample files   ❑ ❑ ❑ ❑
    4.20.3   Online help   ❑ ❑ ❑ ❑
    4.20.4   Printed documentation   ❑ ❑ ❑ ❑
    4.20.5   Online documentation   ❑ ❑ ❑ ❑
    4.20.6   Database   ❑ ❑ ❑ ❑
    4.20.7   Other, please specify   ❑ ❑ ❑ ❑

4.21   Translated documents
    4.21.1   Installation guides   ❑ ❑ ❑ ❑
    4.21.2   Getting started guides   ❑ ❑ ❑ ❑
    4.21.3   Operating manuals   ❑ ❑ ❑ ❑
    4.21.4   Registration cards   ❑ ❑ ❑ ❑
    4.21.5   Marketing materials   ❑ ❑ ❑ ❑
    4.21.6   Technical documents   ❑ ❑ ❑ ❑
    4.21.7   Other, please specify   ❑ ❑ ❑ ❑

4.22   Testing of localised products
    4.22.1   Use of testing checklist   ❑ ❑ ❑ ❑
    4.22.2   Use of testing scripts   ❑ ❑ ❑ ❑
    4.22.3   Regression test   ❑ ❑ ❑ ❑
    4.22.4   Release test   ❑ ❑ ❑ ❑
    4.22.5   Reviewing of documents   ❑ ❑ ❑ ❑
    4.22.6   Other, please specify   ❑ ❑ ❑ ❑

4.23   Success of localisation
    4.23.1   Developer satisfied with the localised product   ❑ ❑ ❑ ❑
    4.23.2   Developer satisfied with the price-quality ratio   ❑ ❑ ❑ ❑
    4.23.3   Customers satisfied with the localised product   ❑ ❑ ❑ ❑

5   TECHNICAL ISSUES

5.1   The way material is sent
    5.1.1   Paper copy   ❑ ❑ ❑ ❑
    5.1.2   Diskette / compact disk / zip   ❑ ❑ ❑ ❑
    5.1.3   Email   ❑ ❑ ❑ ❑
    5.1.4   Transfer through the Internet   ❑ ❑ ❑ ❑
    5.1.5   Other, please specify   ❑ ❑ ❑ ❑

5.2   Applications
    5.2.1   Windows applications   ❑ ❑ ❑ ❑
    5.2.2   Unix / Linux applications   ❑ ❑ ❑ ❑
    5.2.3   Macintosh applications   ❑ ❑ ❑ ❑
    5.2.4   Platform independent applications (e.g. Java, Tcl/Tk)   ❑ ❑ ❑ ❑
    5.2.5   PDA applications   ❑ ❑ ❑ ❑
    5.2.6   Dynamic web-sites   ❑ ❑ ❑ ❑
    5.2.7   Static web-sites   ❑ ❑ ❑ ❑
    5.2.8   Other, please specify   ❑ ❑ ❑ ❑

5.3   Languages used
    5.3.1   C / C++   ❑ ❑ ❑ ❑
    5.3.2   Delphi   ❑ ❑ ❑ ❑
    5.3.3   Visual Basic   ❑ ❑ ❑ ❑
    5.3.4   Java   ❑ ❑ ❑ ❑
    5.3.5   ASP / PHP   ❑ ❑ ❑ ❑
    5.3.6   HTML   ❑ ❑ ❑ ❑
    5.3.7   Script languages (e.g. Tcl/Tk, Perl)   ❑ ❑ ❑ ❑
    5.3.8   Other, please specify   ❑ ❑ ❑ ❑

5.4   The way language and culture specific information is stored
    5.4.1   Hard-coded   ❑ ❑ ❑ ❑
    5.4.2   Resources   ❑ ❑ ❑ ❑
    5.4.3   Database   ❑ ❑ ❑ ❑
    5.4.4   Language modules   ❑ ❑ ❑ ❑
    5.4.5   Other, please specify   ❑ ❑ ❑ ❑

APPENDIX 1: Questionnaire form for software developers

5.5   Document format
      5.5.1   Word processor format (e.g. DOC, WP)   ❑ ❑ ❑ ❑
      5.5.2   Rich Text Format, RTF   ❑ ❑ ❑ ❑
      5.5.3   Adobe Portable Document Format, PDF   ❑ ❑ ❑ ❑
      5.5.4   Hyper Text Mark-up Language, HTML   ❑ ❑ ❑ ❑
      5.5.5   Structured text (e.g. SGML, XML)   ❑ ❑ ❑ ❑
      5.5.6   Plain text   ❑ ❑ ❑ ❑
      5.5.7   Other, please specify   ❑ ❑ ❑ ❑

5.6   Tools used
      5.6.1   Project management tools   ❑ ❑ ❑ ❑
      5.6.2   Programming environments / compilers   ❑ ❑ ❑ ❑
      5.6.3   Design / CASE tools   ❑ ❑ ❑ ❑
      5.6.4   Test support tools   ❑ ❑ ❑ ❑
      5.6.5   Version control tools   ❑ ❑ ❑ ❑
      5.6.6   Structured text editors (e.g. SGML, XML)   ❑ ❑ ❑ ❑
      5.6.7   HTML editors   ❑ ❑ ❑ ❑
      5.6.8   Help-authoring tools   ❑ ❑ ❑ ❑
      5.6.9   Image processing tools   ❑ ❑ ❑ ❑
      5.6.10   Desktop publishing tools   ❑ ❑ ❑ ❑
      5.6.11   Other, please specify   ❑ ❑ ❑ ❑

5.7   Examples of the tools above _____

5.8   What kind of tools would you like to have? _____

---

6   PROBLEMS AND DEVELOPMENT NEEDS

Please specify which kind of problems you have experienced in the following areas and describe how you have dealt with them.

6.1   Project management _____

6.2   Communication _____

6.3   Human resources _____

6.4   Methods and techniques _____

6.5   Internationalisation _____

6.6   Testing and quality assurance _____

6.7   What are the main problems when making a global software product? _____

6.8   How would you improve your globalisation processes? _____

---

7   EXPECTATIONS FROM UNIVERSITIES

7.1   More training for translating students in
       ❑   Translation / foreign language skills
       ❑   Use of basic applications
       ❑   Localisation
       ❑   Internationalisation
       ❑   Software engineering
       ❑   Information technology in general
       ❑   Culture specific issues
       ❑   Other, please specify _____

7.2   More training for software engineering students in
       ❑   Translation / foreign language skills
       ❑   Use of basic applications
       ❑   Localisation
       ❑   Internationalisation
       ❑   Software engineering
       ❑   Information technology in general
       ❑   Culture specific issues
       ❑   Other, please specify _____

APPENDIX 1: Questionnaire form for software developers

7.3 Educated people, specialised in translation, with the degree of
- ❑ Bachelor
- ❑ Master
- ❑ Licentiate
- ❑ Doctor

7.4 Educated people, specialised in globalisation, with the degree of
- ❑ Bachelor
- ❑ Master
- ❑ Licentiate
- ❑ Doctor

7.5 Technology transfer
- ❑ New methods, techniques and models
- ❑ Consultation
- ❑ Training for practitioners

7.6 New tools
- ❑ Ready to use tools
- ❑ Prototypes

7.7 Proposals for research topics _____
_____
_____
_____
_____

8 CLOSING REMARKS

8.1 How well do you know the books in list 1?
8.1.1 I have read the book and found it useful _____
8.1.2 I have read the book but did not find it useful_____
8.1.3 I have heard about the book but I have not read it_____
8.1.4 I do not know the book at all _____

8.2 What are the most important factors when choosing a localisation vendor?_____
_____
_____

8.3 Experiences about localisation vendors _____
_____
_____

8.4 Would you be interested in participating in globalisation research and training with the university?
- ❑ Yes _____
- ❑ No

8.5 Would you like to receive the results of the survey?
- ❑ Yes _____
- ❑ No

8.6 Other comments and suggestions _____
_____
_____
_____
_____
_____

LIST 1. Books

1. *CJKV Information Processing*, K. Lunde
2. *Global Software : Developing Applications for the International Market*, D. Taylor
3. *Global Solutions for Multilingual Applications: Real-World Techniques for Developers and Designers*, C. Ott
4. *International User Interfaces*, J. Nielsen & E.M. Del Galdo (editors)
5. *Internationalization with Visual Basic*, M.S. Kaplan
6. *Internationalization: Developing Software for Global Markets*, T. Luong, J. Lok, D. Taylor & K. Driscoll
7. *Java Internationalization,* A. Deitsch & D. Czarnecki
8. *Programming for the World - A Guide To Internationalization*, S.M. O'Donnell
9. *Software Internationalization and Localization*, E. Uren, R. Howard & T. Perinotti
10. *XML Internationalization and Localization*, Y. Savourel
11. Other, please specify _____

APPENDIX 2: Questionnaire form for localisation vendors

Software Globalisation in Finland: A State-of-the-practice Survey
Questionnaire form for localisation vendors

1 GENERAL INFORMATION

1.1   Place _____
1.2   Time _____
1.3   Participant _____
1.4   Occupation _____
1.5   Job description
      ❑   Management
      ❑   Translation
      ❑   Localisation engineering
      ❑   Other, please specify _____

2 BACKGROUND INFORMATION OF THE COMPANY

2.1   Company name_____
2.2   Home country_____
2.3   Number of offices in Finland _____
2.4   Number of offices abroad_____
2.5   Year established _____
2.6   Net revenue_____
2.7   Number of translation projects_____
2.8   Number of localisation projects _____
2.9   Number of supported languages_____
2.10 Typical supported languages _____
      _____
2.11 Services
      ❑   Translation
      ❑   Proofreading / reviewing
      ❑   Desktop publishing
      ❑   Software localisation
      ❑   Web localisation
      ❑   Documentation / technical writing
      ❑   Testing / quality assurance
      ❑   Consultation
      ❑   Education
      ❑   Other, please specify _____
2.12 Fields of translation specialisation
      ❑   Software
      ❑   Information technology
      ❑   Commerce / finance
      ❑   Marketing
      ❑   Law
      ❑   Medical
      ❑   Engineering
      ❑   Government
      ❑   Literature
      ❑   Other, please specify _____
2.13 Organisational approach to localisation
      ❑   Centralised
      ❑   Decentralised

3 PERSONNEL

3.1   Number of personnel (total and in Finland) _____
3.2   Number of freelance translators (total and in Finland) _____
3.3   Number of localisation personnel (total and in Finland)_____
3.4   Localisation team structure, specialised people in
      ❑   Project management
      ❑   Localisation engineering
      ❑   Translation
      ❑   Technical writing
      ❑   Testing
      ❑   Cultural issues
      ❑   Usability
      ❑   No specialised job descriptions
      ❑   Other, please specify _____
3.5   Typical personnel education / degrees _____
3.6   In-house training per year _____

3.7   Subjects of training sessions _____

3.8   Computing skills of the personnel _____

3.9   Need for new personnel within 3 years (number and fields of specialisation) _____
       _____

4   LOCALISATION PROJECTS

4.1   Sizes of projects in person weeks/months (typical and range) _____
4.2   Duration in weeks/months (typical and range) _____
4.3   Number of translated words (typical and range) _____
4.4   Typical native language of products _____
4.5   Target languages _____
       _____
4.6   Workflow of a typical localisation project_____
       _____
       _____
       _____

1=not applicable / do not know / no comment, 2=never, 3=occasionally, 4=often

4.7	Product type		1	2	3	4
	4.7.1	Software application	❑	❑	❑	❑
	4.7.2	Web-site	❑	❑	❑	❑
	4.7.3	Multimedia product	❑	❑	❑	❑
	4.7.4	Other, please specify	❑	❑	❑	❑

4.8	Components of localisation projects					
	4.8.1	Application	❑	❑	❑	❑
	4.8.2	Sample files	❑	❑	❑	❑
	4.8.3	Online help	❑	❑	❑	❑
	4.8.4	Printed documentation	❑	❑	❑	❑
	4.8.5	Online documentation	❑	❑	❑	❑
	4.8.6	Database	❑	❑	❑	❑
	4.8.7	Multimedia	❑	❑	❑	❑
	4.8.8	Other, please specify	❑	❑	❑	❑

4.9	Translated documents					
	4.9.1	Installation guides	❑	❑	❑	❑
	4.9.2	Getting started guides	❑	❑	❑	❑
	4.9.3	Operating manuals	❑	❑	❑	❑
	4.9.4	Registration cards	❑	❑	❑	❑
	4.9.5	Marketing materials	❑	❑	❑	❑
	4.9.6	Technical documents	❑	❑	❑	❑
	4.9.7	Other, please specify	❑	❑	❑	❑

4.10	Project management					
	4.10.1	Documentation of requirements	❑	❑	❑	❑
	4.10.2	Definition of tasks and activities	❑	❑	❑	❑
	4.10.3	Creation of project schedule	❑	❑	❑	❑
	4.10.4	Creation of project budget	❑	❑	❑	❑
	4.10.5	Allocation of resources	❑	❑	❑	❑
	4.10.6	Monitoring of progress	❑	❑	❑	❑
	4.10.7	Creation of change management procedures	❑	❑	❑	❑
	4.10.8	Identification and documentation of risks	❑	❑	❑	❑
	4.10.9	Other, please specify	❑	❑	❑	❑

4.11	Preparation tasks					
	4.11.1	Use of preparation checklist	❑	❑	❑	❑
	4.11.2	Selection of tools	❑	❑	❑	❑
	4.11.3	Creation of product specific glossary	❑	❑	❑	❑
	4.11.4	Test-compilation of source material	❑	❑	❑	❑
	4.11.5	Use of pseudo translation	❑	❑	❑	❑
	4.11.6	Creation of localisation kit for translators	❑	❑	❑	❑
	4.11.7	Other, please specify	❑	❑	❑	❑

4.12	Localisation engineering tasks					
	4.12.1	Reuse of previous translations	❑	❑	❑	❑
	4.12.2	Use of version control	❑	❑	❑	❑
	4.12.3	Editing of source code	❑	❑	❑	❑
	4.12.4	Customisation of functionality	❑	❑	❑	❑
	4.12.5	Customisation of data	❑	❑	❑	❑
	4.12.6	Other, please specify	❑	❑	❑	❑

APPENDIX 2: Questionnaire form for localisation vendors

4.13   User interface tasks
     4.13.1   Use of UI glossaries   ❏ ❏ ❏ ❏
     4.13.2   Modification of UI   ❏ ❏ ❏ ❏
     4.13.3   Resizing of UI elements   ❏ ❏ ❏ ❏
     4.13.4   Customisation of colours   ❏ ❏ ❏ ❏
     4.13.5   Customisation of font types and sizes   ❏ ❏ ❏ ❏
     4.13.6   Customisation of accelerators   ❏ ❏ ❏ ❏
     4.13.7   Editing of images   ❏ ❏ ❏ ❏
     4.13.8   Replacing of culture specific icons   ❏ ❏ ❏ ❏
     4.13.9   Other, please specify   ❏ ❏ ❏ ❏

4.14   Testing and quality assurance tasks
     4.14.1   Use of testing checklist   ❏ ❏ ❏ ❏
     4.14.2   Use of testing scripts   ❏ ❏ ❏ ❏
     4.14.3   Linguistic test   ❏ ❏ ❏ ❏
     4.14.4   UI validation test   ❏ ❏ ❏ ❏
     4.14.5   Functionality test   ❏ ❏ ❏ ❏
     4.14.6   Interoperability test   ❏ ❏ ❏ ❏
     4.14.7   Delivery test   ❏ ❏ ❏ ❏
     4.14.8   Writing of problem reports   ❏ ❏ ❏ ❏
     4.14.9   Proofreading   ❏ ❏ ❏ ❏
     4.14.10   Other, please specify   ❏ ❏ ❏ ❏

4.15   Success of localisation projects
     4.15.1   Project finishes on-time   ❏ ❏ ❏ ❏
     4.15.2   Project finishes on-budget   ❏ ❏ ❏ ❏
     4.15.3   Customers satisfied with the result   ❏ ❏ ❏ ❏
     4.15.4   Customers satisfied with the price-quality ratio   ❏ ❏ ❏ ❏

5   TECHNICAL ISSUES

5.1   The way material is received
     5.1.1   Paper copy   ❏ ❏ ❏ ❏
     5.1.2   Diskette / compact disk / zip   ❏ ❏ ❏ ❏
     5.1.3   Email   ❏ ❏ ❏ ❏
     5.1.4   Transfer through the Internet   ❏ ❏ ❏ ❏
     5.1.5   Other, please specify   ❏ ❏ ❏ ❏

5.2   Platforms and environments
     5.2.1   Windows applications   ❏ ❏ ❏ ❏
     5.2.2   Unix / Linux applications   ❏ ❏ ❏ ❏
     5.2.3   Macintosh applications   ❏ ❏ ❏ ❏
     5.2.4   Platform independent applications (e.g. Java, Tcl/Tk)   ❏ ❏ ❏ ❏
     5.2.5   PDA applications   ❏ ❏ ❏ ❏
     5.2.6   Dynamic web-sites   ❏ ❏ ❏ ❏
     5.2.7   Static web-sites   ❏ ❏ ❏ ❏
     5.2.8   Other, please specify   ❏ ❏ ❏ ❏

5.3   Types of software localisation
     5.3.1   Source code localisation   ❏ ❏ ❏ ❏
     5.3.2   Text-only resource files   ❏ ❏ ❏ ❏
     5.3.3   Binary localisation   ❏ ❏ ❏ ❏
     5.3.4   Language modules   ❏ ❏ ❏ ❏
     5.3.5   Other, please specify   ❏ ❏ ❏ ❏

5.4   Document format
     5.4.1   Word processor format (e.g. DOC, WP)   ❏ ❏ ❏ ❏
     5.4.2   Rich Text Format, RTF   ❏ ❏ ❏ ❏
     5.4.3   Adobe Portable Document Format, PDF   ❏ ❏ ❏ ❏
     5.4.4   Hyper Text Mark-up Language, HTML   ❏ ❏ ❏ ❏
     5.4.5   Structured text (e.g. SGML, XML)   ❏ ❏ ❏ ❏
     5.4.6   Plain text   ❏ ❏ ❏ ❏
     5.4.7   Other, please specify   ❏ ❏ ❏ ❏

5.5   Tools used
     5.5.1   Project management tools   ❏ ❏ ❏ ❏
     5.5.2   Localisation tools   ❏ ❏ ❏ ❏
     5.5.3   Programming environments   ❏ ❏ ❏ ❏
     5.5.4   Structured text editors (e.g. SGML, XML)   ❏ ❏ ❏ ❏
     5.5.5   HTML editors   ❏ ❏ ❏ ❏
     5.5.6   Help-authoring tools   ❏ ❏ ❏ ❏
     5.5.7   Image processing tools   ❏ ❏ ❏ ❏
     5.5.8   Desktop publishing tools   ❏ ❏ ❏ ❏
     5.5.9   Test support tools   ❏ ❏ ❏ ❏
     5.5.10   Version control tools   ❏ ❏ ❏ ❏
     5.5.11   Translation memory tools   ❏ ❏ ❏ ❏

APPENDIX 2: Questionnaire form for localisation vendors

5.5.12	Machine translation tools	❑ ❑ ❑ ❑
5.5.13	Terminology management tools	❑ ❑ ❑ ❑
5.5.14	Concordance tools	❑ ❑ ❑ ❑
5.5.15	Other, please specify	❑ ❑ ❑ ❑

5.6   Examples of the tools above _____
_____
_____

5.7   What kind of tools would you like to have? _____
_____

6   PROBLEMS AND DEVELOPMENT NEEDS

Please specify which kind of problems you have experienced in the following areas and describe how you have dealt with them.

6.1   Project management _____
_____

6.2   Communication _____
_____

6.3   Human resources _____
_____

6.4   Methods and techniques _____
_____

6.5   Tools _____
_____

6.6   Translation _____
_____

6.7   Localisation engineering _____
_____

6.8   Testing and quality assurance _____
_____

6.9   What are the main problems in localisation industry?_____
_____

6.10   What are the reasons for budget overruns? _____
_____

6.11   What are the reasons for schedule delays?_____
_____

6.12   How would you improve your localisation processes? _____
_____

7   EXPECTATIONS FROM UNIVERSITIES

7.1   More training for translating students in
        ❑   Translation / foreign language skills
        ❑   Use of basic applications
        ❑   Localisation
        ❑   Internationalisation
        ❑   Software engineering
        ❑   Information technology in general
        ❑   Culture specific issues
        ❑   Other, please specify _____
7.2   More training for software engineering students in
        ❑   Translation / foreign language skills
        ❑   Use of basic applications
        ❑   Localisation
        ❑   Internationalisation
        ❑   Software engineering
        ❑   Information technology in general
        ❑   Culture specific issues
        ❑   Other, please specify _____

7.3   Educated people, specialised in translation, with the degree of
- ❑ Bachelor
- ❑ Master
- ❑ Licentiate
- ❑ Doctor

7.4   Educated people, specialised in globalisation, with the degree of
- ❑ Bachelor
- ❑ Master
- ❑ Licentiate
- ❑ Doctor

7.5   Technology transfer
- ❑ New methods, techniques and models
- ❑ Consultation
- ❑ Training for practitioners

7.6   New tools
- ❑ Ready to use tools
- ❑ Prototypes

7.7   Proposals for research topics _____
_____
_____
_____
_____
_____

8   CLOSING REMARKS

8.1   How well do you know the books in list 1?
8.1.1 I have read the book and found it useful _____
8.1.2 I have read the book but did not find it useful _____
8.1.3 I have heard about the book but I have not read it_____
8.1.4 I do not know the book at all _____

8.2   Would you be interested in participating in globalisation research and training with the university?
- ❑ Yes _____
- ❑ No

8.3   Would you like to receive the results of the survey?
- ❑ Yes _____
- ❑ No

8.4   Other comments and suggestions _____
_____
_____
_____
_____
_____

LIST 1. Books

1. *A Practical Guide to Localization*, B. Esselink
2. *CJKV Information Processing*, K. Lunde
3. *Global Solutions for Multilingual Applications: Real-World Techniques for Developers and Designers*, C. Ott
4. *Guide to Macintosh Software Localization*, T. Luong, J. Lok, D. Taylor & K. Driscoll
5. *International Technical Communication : How to Export Information About High Technology*, N.L. Hoft
6. *International User Interfaces*, J. Nielsen & E.M. Del Galdo (editors)
7. *Software Internationalization and Localization*, E. Uren, R. Howard & T. Perinotti
8. *XML Internationalization and Localization*, Y. Savourel
9. Other, please specify _____