Empirically Researching Development of International Software

Malte Ressin, José Abdelnour-Nocera, Stephen Roberts
University of West London
Centre for Internationalisation and Usability
London, UK
e-mail: malte.ressin/jose.abdelnour-nocera/stephen.roberts@uwl.ac.uk

Abstract—Software localization is an important process for international acceptance of software products. However, software development and localization don’t always come together without friction. In our empirical software engineering research, we examine the interplay of software development and software localization by gathering and analyzing qualitative and quantitative data from professionals in relevant roles. Our aim is to co-validate issues and inform practice about the development of international software.

Keywords—empirical software engineering; industrial practice and experience

I. INTRODUCTION

To be successful in international markets, software products should be adapted to the intended target markets [1]. This adaption’s importance remains high since the number of cultures a product will get in touch with increases with continuing globalization [2], and because many modern software innovations emphasize human factors [3] such as expectations and perceptions, and therefore are related to cultural factors. If adapted well, the software has good chances of retaining its original usability [4]. Otherwise, it might become increasingly difficult to use [5].

The process of adapting is called localization [6] and mostly covers translation of text, units and formats, content changes, colors, layout, but possibly also functionality and provisions for technical infrastructure [7]. Localization is often preceded by internationalization, i.e. provision of a culturally neutral version which can be “filled” with content dependent on specific target markets [8]. Localization is often done after it has been internationalized by software engineers [9] by translators who had no input into the product’s development [10].

Although the cost and effort for localization can be considerable [11], no standard workflows exist [12] and approaches how to bring together software development and localization processes have not been agreed upon [4]: Some find reasons to integrate the latter into the former [13], while others advocate a strict separation [14]. Academic literature has so far focused on tools, technologies and activities of individuals, while workflows across all disciplines go mostly unexamined [12]. However, there are indications that it is the collaboration of software developers and translators, and the interplay of their different workflows and working styles, that lead to issues in software localization [15].

This research examines the development process of international software by exploring the work of professionals in various roles, why they work that way, and the resulting implications, how localization activity is connected to development activity, and how localizers interact with developers. We gather statistical data about software localization work and interdisciplinary differences to support our qualitative findings.

Our aim is to provide a better understanding of the process of software localization and interdisciplinary development of international software. This understanding will serve to inform the practice of involved roles, e.g. developers, project managers and translators, in order to facilitate development of international software and mitigation of localization issues.

II. LOCALIZATION IN SOFTWARE DEVELOPMENT

Cost, time, and quality are the main issues in software localization [1]. Cost refers to any direct or indirect additional expenditure through the localization effort. Time refers to the delay of a product release through software localization. Quality refers to localization completeness, appropriateness, comprehensibility etc.

These issues can have causes in the interplay of software development and localization [16], and mismatches between development methodologies and localization work have been documented [17]. Human factors, i.e. the individual work of developers and translators and their collaboration in the development of international software, can also be disruptive: During development, localization is easily reduced to translation and language mechanics (e.g. [18]), and stock solutions for multi-language support are part of many programming frameworks and programming languages [17]. Beyond these engineering requirements, which can be precisely formulated, localization has also emotional requirements, which are vague and more difficult to convey [19], and pose specific challenges to requirements engineering [20]. As a result of emotional requirements, translators for example require context documentation [6], a requirement often disregarded in software development.
III. MODEL OF SOFTWARE LOCALIZATION ASPECTS

It is impractical to base an understanding of software localization solely on the three issues cost, time and quality. We therefore approach software localization following a different model. Initial literature research and written reports, discussions with professionals and researchers, and an analysis of our own practical experience, made us consider the topic under the following three aspects:

- **Infrastructure**: tools and how they are used.
- **Processes**: the intersections of different work steps by developers and localizers.
- **Human factors**: perception, knowledge and attitude of relevant roles in the development of international software.

These three aspects have only blurred borders and overlap. For example, communication has a tool aspect, i.e. whether it is done by phone, email or in person. It also has a procedural aspect, depending on it being sporadic or as part of a procedure. Last but not least, it has various human aspects such as the opportunity to misunderstanding caused by differences in knowledge and expectation.

We intend to examine software localization based on facts and co-validate issues in order to gain a better understanding of software localization and to ultimately inform practice. Our fundamental research questions are: What influences do the three aspects human, processes and infrastructure have on the three issues cost, time and quality? How are cost, time and quality determined by the three aspects?

We divide the exploration of this model into three phases:

1) **Interviews (Type A)** with professionals who have worked in the development of international software from different roles.

2) **On-going interviews (Type A)** and a parallel quantitative data collection via a web-based, cross-sectional survey. This survey is influenced by early interviews and queries software localization infrastructure, processes, and perception, knowledge and communication.

3) **Interviews (Type B)** with professionals from different roles, informed by results of previous research.

In the Interviews Type A in phase 1, we gather data by exploratory interviews, either face-to-face or by telephone. Participants of different roles relate the properties of their projects, their processes, their experiences, etc. These aspects are approached by looking at current software development and localization practices, and the occurrence of the three localization issues. Since we examine development and localization practices and issues from different roles, we’re able to form conclusions about interdisciplinary collaboration in practice, the motivations for processes and methods, how they are perceived by different disciplines, and how localization issues occur.

Interviews are analyzed bottom-up to discover recurring themes and issues, using thematic analysis, framework analysis and technological frames [21].

In phase 2, while interviewing continues, a web-based cross-sectional survey is conducted. This survey is informed by the results from initial research and the first interviews. The survey tests localization issues reported in the literature, e.g. alleged indifference of developers toward localization [16]. It further queries infrastructure, e.g. available localization tools, processes, e.g. specific work steps, and human factors, e.g. self-efficacy (one’s perception of being able to affect change) in software localization. It contains a scale measuring attitude towards software localization, based on the Attitude Towards Computer Technology (ACT) instrument by Kinzie et al. [22], and further contains the Cultural Intelligence Scale [2] to measure cultural competence. The survey supplements the interview data for the three localization aspects with quantitative data:

- **Human factors**: Cultural competence and its relation to localization effort; attitude towards localization and its relation to development role; self-efficacy in localization and its relation to cultural competence and attitude.
- **Infrastructure**: Statistical data about tool availability and use; localization effort and its relation to product type; localization importance and tool availability.
- **Processes**: Statistical data about processes; localization process and its relation to product type, localization importance and tool availability.

In addition, the survey measures simple localization quality criteria, for example which software parts are localized (text, color, functionality, etc.), whether quality assurance processes are in place, etc. The relationship between these and the three factors mentioned above is examined as well. Survey results are analysed for factor loading, regression tests and analysis of variance (ANOVA).

Phase 3 continues with structured interviews (Type B) based on the findings from the previous two phases, and focus group interviews, separated into groups of professionals from the technical development domain, the linguistic domain, and a combination of the two. Analysis again employs thematic analysis, framework analysis and technological frames.

The results of qualitative data gathering, particularly accounts during interviews, lead us to interrelations between the three above-mentioned localization aspects infrastructure, processes and human factors, with the three localization issues cost, time and quality. Results of quantitative data gathering lead us to collateral interrelations between two localization issues cost and quality.

IV. PROGRESS

We are nearing the end of phase 2 of our research: Based on the initial research idea, we have conducted a literature review to refine our research questions, aims and methods.
One of the results our literature review contributed significantly to is the partition of software localization into its three aspects infrastructure, processes and human factors. In addition to the review of academic literature, we have reviewed professional reports on software localization.

An early short survey was run between 18 and 20 April 2011 at the 19th International Software Quality Management conference in Loughborough, UK. Unfortunately, the response rate of this survey was too small for scientifically reliable results. However, we were able to gather valuable experience, gained interesting contacts, and suggestions for further research.

We have conducted number of interviews with professionals from the following roles:

- Three project managers
- One UI designer
- Three software engineers
- Two linguists

Based on literature research, experience from the short survey, and the interviews, we have developed the survey in the summer of 2011. It was piloted in early October 2011 with ten software localization professionals. After a few minimal changes to phrasing and question order, it was published on 20 October 2011. Currently, we have ca. 40 completed surveys from a total number of ca. 70.

V. REMAINING WORK

We continue advertising our survey incrementally until mid-April 2012 and keep it open until at least 30 April 2012. Our response target is to have 30 completed surveys from each of the roles software developer, project manager, localizer, and UI designer.

As outlined in our research approach, we continue to interview professionals from international software development, ideally bringing up the number of interviewed individuals from each of the following roles (project manager, software developer, localizer/translator, UI specialist) to at least five in the post-survey interviews (Type B), and by conducting focus group interviews.

We are very confident to reach the targeted participation numbers. Regarding the survey, while we have already advertised the survey in different media and venues, there are still a number of channels and contacts left that have been identified, but not tapped in yet. In addition, several survey participants agreed to be interviewed and we have already more interviews scheduled.

VI. CONCLUSIONS AND LIMITATIONS

We are aware of the different epistemological positions we are taking in this research: On one hand, there is a very strong constructionist aspect through the method of self-reporting perceptions and interpretations of our participant. This is particularly important because some of the phenomena we’re examining, for example human factors such as collaboration and individual contribution, but also infrastructure and process aspects like usage of tools and implementation of processes, are not only self-reported, but are also an interpretation of the reporter and do not exist objectively in itself.

On the other hand, we’re also looking at objectively existing artifacts such as availability of specific tools, objectively quantifiable plans and errors, etc. While still self-reported, these are objectively existing and less dependent on individual interpretation and as such carry a positivistic trait.

This use of different epistemological positions allows us to use interview and survey data to triangulate the findings. However, our findings remain unevaluated, and our aim is not to create a model predicting outcomes or implying causality. Instead, we intend to characterize issues and inform practice. Our contribution to improving software localization practice, in accordance to critical theory, lies in aiding professionals in interpreting their experiences during software localization towards understanding the causes, and improving their practices by aligning processes, tools, requirements, and expectations.

VII. PUBLICATIONS


REFERENCES