Decoding Software Localisation Practice

School of Computing and Engineering

Malte Ressin

Supervisors: José Abdelnour-Nocera, Stephen Roberts

Contact e-mail: malte.ressin@uwl.ac.uk

UNIVERSITY OF WEST LONDON

Abstract

Software localisation, the adaptation of software products for different languages, cultures and markets, is an important activity for international software. However, reports suggests that work of developers and translators does not mesh seamlessly, causing disproportionate cost, lack of quality, and delayed product release. Yet there is little research on localisation or its human factors. This research examines the causes of localisation issues by analysing qualitative data about the collaboration between development and translation. Semi-structured interviews with professionals in various roles were analysed towards a grounded theory of interdisciplinary collaboration in software localisation, explaining how collaboration strategies and conflicts reciprocally affect each other and are affected by external influences. Results suggest gaps in knowledge, procedure and motivation between developers and translators, as well as a lack of cross-disciplinary knowledge and coordination.

Introduction

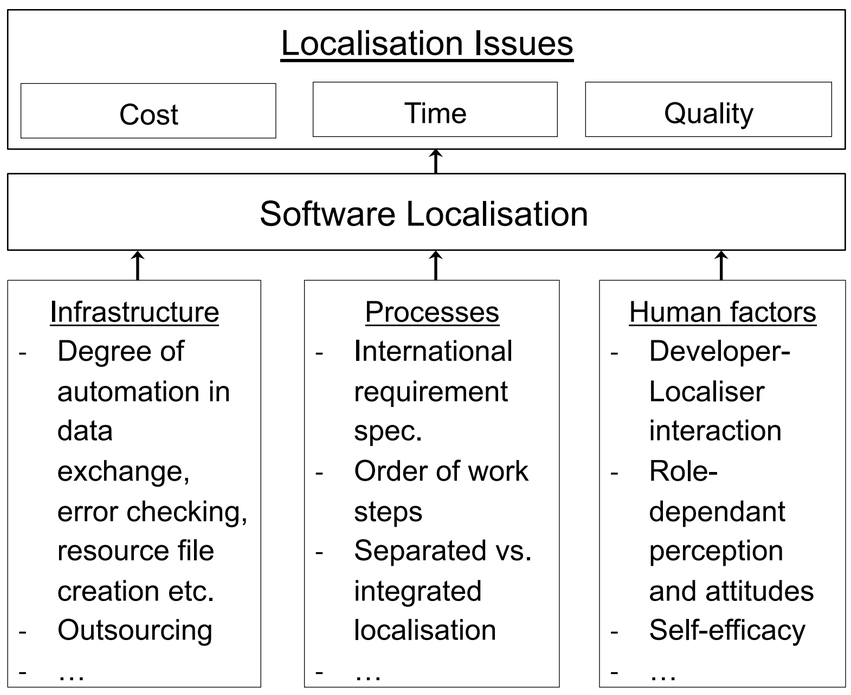
This research started as a puzzle from my work as localisation team leader in a mid-sized software company: I had observed that most localisation came from seemingly trivial causes nonetheless defying any attempts at proactive prevention. These could usually be phrased in the form of if-only, for example:

- If only software engineers finalised user interface (UI) text a month before product release, there would be no translation-caused release delays.
- If only UI designers remembered to leave at least 30% buffer space for translation- expanded text, there would be fewer instances of cut text in the UI.
- If only translators referred to the terminology when translating, we would have fewer retranslations.

Process-related shortcomings in localisation practice have further been acknowledged in the literatur:

- Lack of standard processes (Abufardeh and Magel, 2008)
- Incomplete understanding of localisation activities and workflow (Lenker et al., 2011)
- Issues of collaboration between software engineering and localisation (Abufardeh and Magel, 2010; Lewis et al., 2009).

Accordingly, there have been calls to examine the collaboration of software engineering and localisation (O'Sullivan, 2001; Collins, 2001). The conceptual model guiding initial research was based on the project management triangle of cost, time and quality.



Aims and Objectives

For this research, the following aims and objectives were defined:

- Analyse accounts about localisation practice, in particular regarding the cooperation of developers and localisers.
- Examine the role of human factors in localisation as a process.

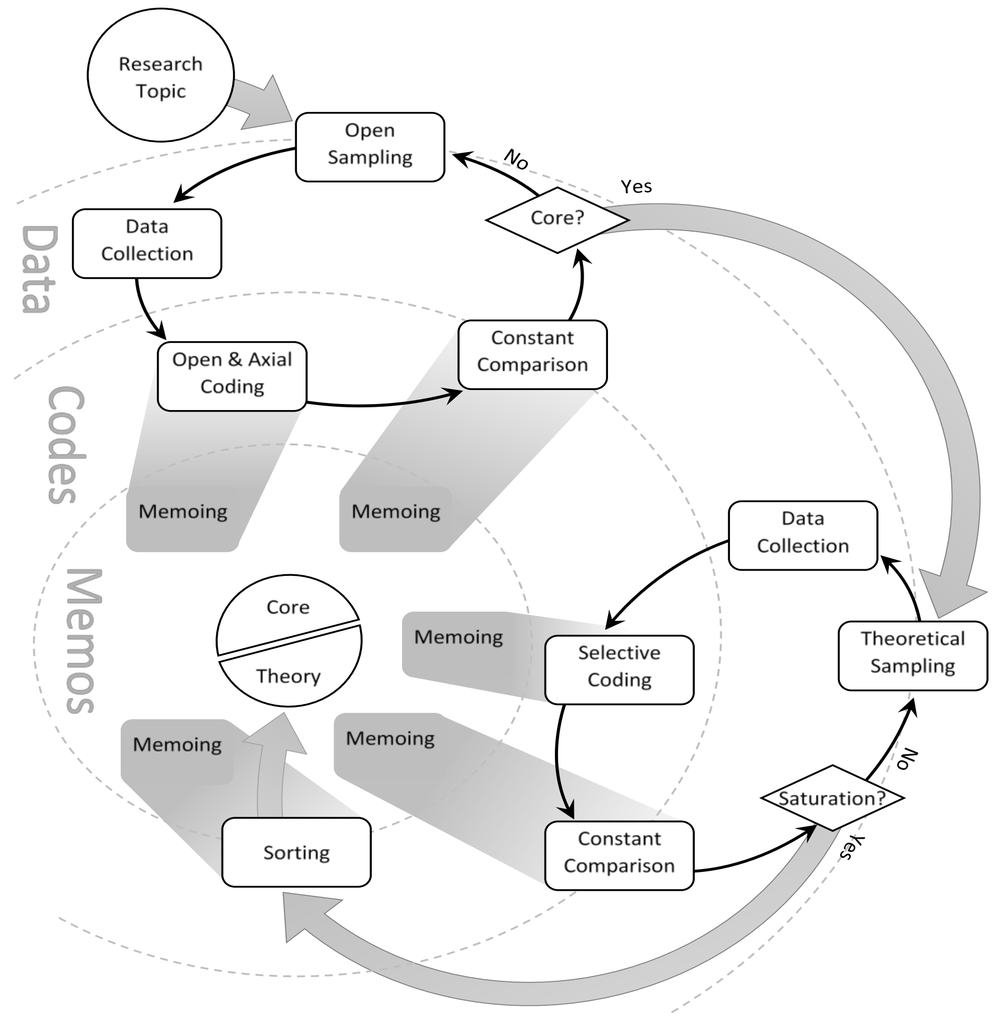
Based on these, two research questions are formulated:

RQ 1 How is localisation conducted individually and collaboratively by developers and localisers, and how does this shape each discipline's activities?

RQ 2 How are issues caused during localisation and internationalisation?

Methodology

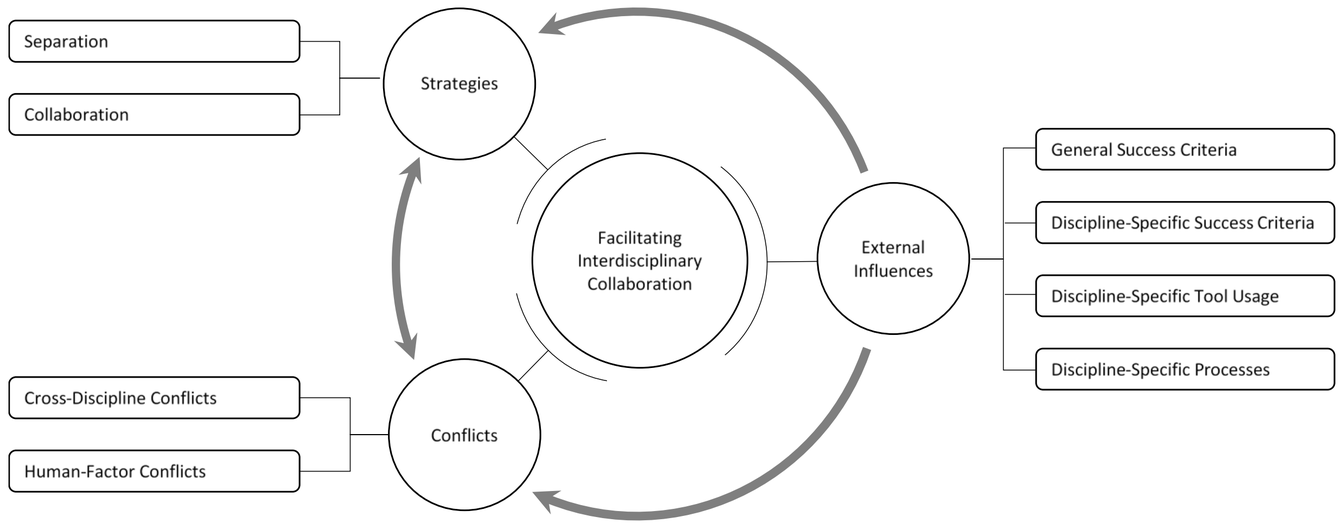
The literature review suggested that software localisation is conducted in a social context of software development and is affected by human factors. For such phenomena which are difficult to study in isolation, qualitative research is the appropriate approach. Interviews were chosen for this research because they enable the collection of both accounts of practice and of insights into participants' thoughts and opinions, and have the potential to reveal what happened, how it happened, and why it happened in that way. Compared to observation, the data amount in interviews is moderate and analysis affordable. To examine how localisation is conducted, how this shapes the activities of localisation and internationalisation, and how it causes localisation issues, Straussian Grounded Theory is used: A participant for an interview, is selected based on availability. Ideally, cases are selected to broadly cover the research subject. Data is gathered by interviewing a participant, and then coded using open coding and axial coding. During open coding, all the data is scrutinised for concepts through line-by-line coding. In axial coding, the concepts gathered in open coding are examined for dimensions which relate to their occurrence, e.g. causal and intervening conditions, context and consequences. All the concepts appearing in the data are coded. The codes are post-formed, i.e. they are derived from the data. The aim is to find a structure relating concepts.



Once coding of a case finishes, the next case is selected, data is gathered and analysed, and so on. Any theory derived from a new case is retroactively applied to all previous cases. This is referred to as constant comparison, meaning that information from incoming data is constantly compared to previously analysed data. New codes, derived from the latest case during open coding and thus post-formed, are applied to previous cases, and thus become preformed. The researcher notes any theoretical insights gained during data analysis in so-called memos. This process of selecting and analysing a case, constantly comparing new and previous cases and writing memos is repeated until a single core category emerges.

Results and Conclusions

A grounded theory emerged bottom-up from processed, reduced and organised interview data. Below is an overview of the theory of interdisciplinary collaboration and main answers to the research questions. Circles signify categories, boxes signify high- level concepts, lines indicate relationships, and arrows indicate influences.



What shapes the activities of developers and localisers?

A range of processes are used to localise software. The work of developers and localisers is strongly influenced by the strategic choice of conducting localisation in-house or out-of-house. Activities are shaped by external influences such as success criteria, limitations and affordances of tools, and limitations and tasks prescribed in the overall organisation. Additionally, the activities are modified to avoid or handle conflicts, i.e. failures and impasses, previously experienced or expected in localisation.

What causes localisation issues?

Localisation issues are a results of the hierarchical relationship between developers and localisers. Developers enjoy a privileged position compared to localisers and their relationship with localisers can easily develop into a dysfunctional regime in which processes and tools exclusively cater for development,. The more dysfunctional the relationship is, the less localisers request necessary information or warn about potential issues, and instead shift their work and activities towards their unique interests. Eventually, developers and localisers settle into a relationship in which localisation is superseded by alternative interests. Those goals of software localisation which are not among their priorities are compromised and cost, quality or schedule issues occur.

References

Abufardeh, S. and Magel, K. (2008). Software localization: the challenging aspects of Arabic to the localization process (Arabization). In: *Software Engineering SE 2008*, Innsbruck, Austria, p.275–279.

Abufardeh, S. and Magel, K. (2010). The Impact of Global Software Cultural and Linguistic Aspects on Global Software Development Process: Issues and Challenges. In: *NISS 2010*, Gyeongju, Korea, p. 133–138.

Collins, R. W. (2001). Software Localization: Issues and Methods. In: *European Conference on Information Systems ECIS 2001 Proceedings*, Bled, Slovenia, p.36–44.

Lenker, M., Anastasiou, D. and Buckley, J. (2011). Workflow Specification for Enterprise Localisation. *Localisation Focus*, 9 (1), p.26–35.

Lewis, D., Curran, S., Feeney, K., Etzioni, Z., Keeney, J., Way, A. and Schäler, R. (2009). Web Service Integration for Next Generation Localisation. In: *SETQA-NLP '09*, Stroudsburg, PA, USA, p.47–55.

O'Sullivan, P. (2001). Pat O'Sullivan's award winning localisation thesis covers new ground. *Localisation Ireland*, 5 (2), p.6–10.