UWL REPOSITORY

repository.uwl.ac.uk

Agile culture clash: localisation issues in modern software development

This is the Accepted Version of the final output.

Alternative formats: If you require this document in an alternative format, please contact: open.research@uwl.ac.uk

Agile Culture Clash:
Localisation Issues in Modern Software Development

TVU London
Thames Valley University

Author: Malte Ressin   Supervisors: Dr. José Abdelnour-Nocera, Prof. Andy Smith, Prof. Tony Clark

Abstract:

Modern software development is expected to be both flexible and efficient despite continuously increasing project complexity. Partly in response to these requirements, agile software development models have become more and more popular.

The integration of localization in agile software development, however, leads to a number of conflicts. Translation work is expensive and time-consuming. Usually, neither qualified native nor second-language speakers are available to assist or test during development. The localization efforts are outsourced to external translators. As a consequence, a number of important questions arise: How can linguistic and cultural knowledge be communicated for consideration in agile development? How can the need for deadlines be aligned with a desire for agility? How can external dependencies such as translators be integrated seamlessly?

In this poster, we explore these issues and questions. In order to hint at the wide range of possible issues, a sample of common localization problems are shown and the challenges of dealing with them during agile development are outlined.

Localisation Issue Examples:

String length expansion in the UI:

Source sentence: *German target sentence:*

„Add" → „Hinzufügen"
(~250% increase)

„Please refer to the manual for further information." → „Bitte konsultieren Sie das Handbuch, um weitere Informationen zu erhalten."
(~50% increase)

String length expansion during development can lead to unplanned UI format adaption.

Grammar issues, e.g. with dual plural forms:

Source sentence: *Russian translation:*
„It takes 1 **day**."   „Для этого понадобится 1 **день**."
„It takes 2 **days**."   „Для этого понадобится 2 **дня**."
„It takes 5 **days**."   „Для этого понадобится 5 **дней**."
...   ...

implemented as: *Error!*

```
"It takes 1 day."
"It takes %s days."
```

translated to:

```
"Для этого понадобится 1 день."
"Для этого понадобится %s дня."
```

This example shows English-based implementation which can't display Russian without code changes.

Integrating Localisation into Development:

Localisation in a linear development process:

Specification → Localisation → Development → Release

In a linear development process, localisation can be placed between specification and development.

Localisation in an agile development process:

Specification → Development → Re-scoping → Development → Re-scoping → Release
Localisation → ? ? ? ?

In agile, a localisation step during each development is cost-prohibitive and feedback cycles are difficult.

Localisation:
Software adaption for foreign markets.
• Text translation.
• Appearance changes.
• Functional adaptions.
Strictly *linear* work process, often done by external contractors.

Agile Development:
Development process aimed at low risk.
• Self-organized teams.
• Emphasis on short-term planning.
• Short development cycle.
Highly *iterative* work process, tuned to flexibility.

References:

Asnes, A. (2009), Internationalization and automation. *MultiLingual*, 20 (3), p.22-23.

Smith, A., Dunckley, L., French, T., Minocha, S. & Chang, Y. (2004), A process model for developing usable cross-cultural websites. *Interacting with Computers*, 16, p.63-91.