

Why internationalisation is an opportunity

Rob Sexstone explains how planning for localisation at the outset of your software development can reap benefits.

Introduction

This article describes the important distinction between localisation and internationalisation and argues the case for technical communicators working in a software development team to extend their core skills and influence into localisation management.

The evolving development landscape

As the traditional waterfall approach to software development through analysis, specification and design makes way for 'extreme' and 'agile' practices, more than ever before, *not* having a localisation strategy introduces complexity and risk. Translation and localisation do not naturally fit with agile development, because they traditionally rely on complete contexts and for a product or documentation set to be completed before work begins. Without careful and early consideration in an agile environment, last-minute localisation can lead to overly complex administration and unexpected quality issues.

Bridging the knowledge and experience gap

Because of the dynamic and fast-growing nature of the UK's localisation industry, there is a very stark shortage of experienced and talented localisation specialists and managers. Development teams usually rely on one individual (if they are lucky) who has had some experience of working with a translation agency at some point in the past. In general, developers have a limited appreciation of the commercial risks associated with sending a

string table or UI resource file to a translator, having them translate it and then building a release. The negative impacts of this short-term approach to 'translating' string sets rather than 'localising' typically lead to poor version control, lack of consistency, poor user satisfaction in the target locale leading to unexpected change requests, and unnecessary administrative time spent troubleshooting terminology issues. The key point here is; within your team, appoint a localisation evangelist and encourage them to engage with all functions (for example, marketing, development, tech support) as well as with a reputable localisation service provider, who will provide a blend of tactical and strategic advice, support and experience. As Daljit mentioned in the Autumn 2011 *Communicator* "... the single most useful thing to do when planning for international course design is to involve a localisation partner upstream and very early in the development lifecycle. A reliable and experienced localisation partner will be able to steer your development team towards best practices and avoid the common pitfalls and assumptions many monolingual development managers make."

International software design checklist

When engaging with development managers, at 3di, we begin by interviewing them using a standard checklist to assess the localisation readiness of their software product (as well as the team itself!). This often provokes fruitful discussions and highlights risk at

within your team, appoint a localisation evangelist

Complexity made clear.

Expert documentation and localization services

Technical Communication

- ◉ supply of technical authors
- ◉ managed outsourced teams
- ◉ information & document design
- ◉ modular writing training
- ◉ tools & process strategy
- ◉ project management
- ◉ software usability testing

Localization & Translation

- ◉ software products & online help
- ◉ website & e-learning content
- ◉ interactive & audio content
- ◉ technical & compliance documents
- ◉ multi-lingual translation
- ◉ scalable localization testing
- ◉ process design & consultation



www.3di-info.com

+44 (0)1483 211533 | contact@3di-info.com

3di

Table 1. 3di's international software design checklist: 30 software engineering considerations

Software engineering considerations	Relevance*	Priority*
1. Design team considers translation and localisation from the beginning of the project.		
2. All international editions are compiled from one set of source files.		
3. Localisable items are stored externally in resource files, or resource bundles.		
4. Code supports Unicode or conversion between Unicode and local codepages.		
5. String buffers are large enough to handle translated words and phrases.		
6. No assumptions are made that one character storage element represents one linguistic character.		
7. Validate databases to ensure that schemas, datatypes and table design are ready for a multi-locale environment.		
8. All language editions can deal with one another's data.		
9. Program takes advantage of generic text layout functions when available.		
10. International laws affecting design and operation are considered.		
11. Code uses generic datatypes and generic function prototypes if available.		
12. Program handles input of international data.		
13. Program contains support for locale-specific hardware if required.		
14. The product runs properly in its base language in all locales.		
15. Program depends on operating or runtime system functions for sorting, character typing and string mapping.		
16. Third-party components and software used in the product are examined for I18N support.		
17. Strings are not assembled by concatenation of fragments.		
18. Source code does not contain hard-coded character constants, numeric constants, screen positions, filenames or pathnames that assume a particular language.		
19. Code is generic enough to handle the required range of character sets.		
20. Code properly handles all characters in the program's character set.		
21. Code processes all character sets correctly regardless of character widths.		
22. Application works correctly on localised editions of the target operating system.		
23. Program meets international testing standards.		
24. Icons, cursors and bitmaps are generic, are culturally independent and do not contain text whenever possible.		
25. Code does not use embedded font names or make assumptions about particular fonts being available.		
26. Displayed and printed text uses appropriate fonts.		
27. Menu and dialog-box keyboard assignments are unique.		
28. If ethnocentric graphics, colours or fonts are used, they can be replaced dynamically using locale-sensitive switch statements.		
29. Sorting and case conversion are culturally correct.		
30. Program handles user keyboard layout changes.		

* High / Medium / Low

Table 2. 3di's international software design checklist: 12 international usability, UI and human factors

International usability, UI and human factors	Relevance*	Priority*
1. Consistent terminology is used in messages.		
2. UI language strings are reviewed for meaning and spelling to reduce user confusion and lessen translation errors.		
3. Menus, dialogs and Web layouts can tolerate text expansion.		
4. Strings are documented using comments to provide context for translators.		
5. Users can type all supported characters into documents, dialog boxes and filenames.		
6. Shortcut-key combinations are accessible on all international keyboards.		
7. Program responds to changes in the user's choice of international settings (for example, UI language can be changed through a straightforward menu option).		
8. Translated text meets requirements of end users who are native speakers.		
9. Dialogs and forms are resized and UI text is aesthetically presented.		
10. Translated dialogs, toolbars, status bars and menus fit on the screen at different resolutions.		
11. User can successfully cut, paste, save and print text regardless of language.		
12. Are there established test plans and tools for the source product, and can they be applied to localised versions?		

* High / Medium / Low

an early stage for us as a localisation service provider. By providing a starting point for discussions, we find that engagement with the right stakeholders in a development project happens much quicker, and the relationship between international design and ongoing cost efficiency for all future localisation work is taken seriously very early on. Localisation process management is a broad topic beyond the scope of this article and it is something we shall explore in the future.

See 3di's International Software Design Checklists in Tables 1 and 2.

Summary

As the roles of developers become more agile and customer-oriented, so in turn do the roles available for us as technical communicators. Myriad opportunities exist for the traditional technical communicator, such as quality assurance, usability, technical support and web marketing. Effective translation and localisation management embraces and links all of these functions within a modern development team and demands a high level of technical communication skills. The international success of your software product will not happen by accident, and as professional communicators we have a responsibility to embrace a more international and pluralist approach to our endeavours. As we all observe Western domestic economies stagnating, internationalisation, localisation and export seem more relevant than they have been for generations. 

Rob Sexstone MISTC is head of research and development for 3di Information Solutions in Guildford, Surrey.

E: rob.sexstone@3di-info.com

W: www.3di-info.com

Terminology

Localisation. The translation and adaptation of software or multimedia products for another locale.

Internationalisation. The engineering and design effort behind preparing your software or multimedia product for effective localisation. Internationalisation is a software development activity that benefits from collaboration and planning with a localisation partner or consultant in order to streamline future software localisation processes.

References

- Lunde, Ken PhD *CJKV Information processing*, O'Reilly ISBN 1-56592-224-7
- Deutsch and Czarnecki, *Java Internationalization*, O'Reilly ISBN 0-596-00019-7
- Savourel, Yves *XML Internationalization and Localization*, SAMS ISBN 0-672-32096-7
- Sexstone, R, Hanspal, D (2011) 'Localising e-learning – an overview' *Communicator*, Autumn 2011:48-50

Technical resources for Java and XML developers who are interested in I18N

<http://developers.sun.com/techtopics/global/index.html>