Towards Improving Automation Support for Internationalization and Localization Testing

Maria Raquel Lopes de Couto[1,2], **Breno Miranda**[1]

[1] Informatics Center – Federal University of Pernambuco (UFPE)
Recife – PE – Brazil

[2] Programa de Pós-Graduação em Ciência da Computação - Universidade Federal de Pernambuco

Nível do curso: Mestrado, Data de início: May-21, Data de término: Aug-23

{mrlc,bafm}@cin.ufpe.br

Abstract. *Internationalization (i18n) and localization (l10n) testing are crucial to assure the quality and success of a globalized software since its most common failures are easily noticeable on the user interface for the final users. However, this type of testing is often put aside or performed only at late development stages. Furthermore, there is a lack of studies and empirical evidence on the availability and usage of automated tools for supporting i18n and l10n Testing. This research aims at understanding the state-of-the-art in internationalization and localization testing and, based on our findings, propose and evaluate automated solutions for supporting practitioners.*

Resumo. *Testes de internacionalização (i18n) e de localização (l10n) são cruciais para garantir a qualidade e o sucesso de um software globalizado, pois suas falhas mais comuns são facilmente perceptíveis pelos usuários finais. No entanto, esse tipo de teste geralmente é deixado de lado ou realizado apenas em estágios finais de desenvolvimento. Além disso, faltam estudos e evidências empíricas sobre a disponibilidade e o uso de ferramentas automatizadas para dar suporte aos testes i18n e l10n. Esta pesquisa visa compreender o estado da arte em testes de internacionalização e localização e, com base em nossas descobertas, propor e avaliar soluções automatizadas para apoiar os profissionais.*

1. Research Problem

In order to reach the global market, software companies need to bring Internationalization (i18n) and Localization (l10n) into their products. Such process is more complex than simply translating all the strings from the application. The internationalization process refers to designing the software such that it can handle multiple languages and cultural conventions, whilst localization involves taking a product and making it linguistically and culturally appropriate to a country or region in which the software will be used or sold.

To ensure quality and reinforce that the software meets the conventions and requirements of an specific culture, it is necessary to perform internationalization and localization testing on the software. However, performing i18n and l10n testing is challenging: generally, the testers do not have the knowledge of all tested languages; the testing is not carried out for all supported languages, only for a selected scope of languages; and some companies or developers do not test their applications at all [Awwad and Slany 2016], neither automatically nor manually. Besides that, according to

[Ramler and Hoschek 2017a] localization testing is rarely investigated in scientific work and there are only a few reports on automated approaches for localization testing providing very little empirical results or practical advice.

The Localization and Internationalization issues are sometimes classified as cosmetic issues (bugs that does not affect the functionalities from the software, but are visible in the UI). However, they can be decisive for the success of the software. For example, a common issue from localization is the ellipsis: it happens when a translated string does not fit in the UI and shows continuity with ellipsis. There occurrence of ellipsis can cause two severe issues: the ellipsis can occur in the middle of a word and form a swearword; the unfinished sentence cannot provide the instructions needed for the user to complete an action. An example of this issue is shown in Figure 2 (left) and more examples of i18n/l10n issues are described in Section 3.

Given this scenario, this Master Thesis aims at understanding the state-of-the-art (e.g., common testing techniques and challenges faced by the localization and internationalization testing) and, based on our findings, proposing and evaluating automated solutions for supporting practitioners. The investigation of the state-of-the-art will be done through a systematic literature review (SLR). Once we have the results from the planned SLR we aim at proposing a tool to support i18n and l10n testers into performing their tasks automatically. As main contributions of this work, we expect to deliver i) a document that can be used as guideline by i18n and l10n testing professionals; and ii) a tool for supporting automated i18n and l10n testing.

The remainder of this Master thesis proposal is organized as follows: Section 2 describes the main concepts of this work. Section 3 presents the context and motivation of this master thesis. In Section 4 we present the proposed methodology to the execution of this work. Section 5 displays the results obtained so far along with the proposed methodology. Finally, in Sections 6 and 7, we discuss the related work and expected results.

2. Background

The *Globalization* (g11n) of a software is done through localization and internationalization. It can be defined as the process of a company launching its products in the international market [Štefan Molnár 2016].

Internationalization, in turn, is more related to the code design and it is a crucial step for globalization and localization process. It is responsible for preparing the software to adapt to different languages, regions and cultures. By doing the internationalization of the software, it should be possible to display appropriate date format, characters, numbers, and measure units for an specific locale [Ynion 2020]. A *locale* can be defined as the combination of a language and a region, for example, the following two locales: pt-BR and pt-PT. Portuguese (pt) is a language, which is spoken in both Brazil (BR) and Portugal (PT), and each locale (pt-BR and pt-PT) accommodates the particularities about the Portuguese language in each country/culture.

Usually, the applications are developed originally in English, so it is the source language. To adapt the software for localization there are a few good practices recommended, such as: externalizing the resources; making the layout more flexible for adjusting the User Interface (UI) for longer translations; changing the code to adhere to

different locale formats (names, dates, units and so on); and having bidirectional support for Right-to-left locales (locales that the reading is made from the right to the left). Figure 1 displays an example of the i18n process in practice. Instead of using hard-coded strings for "Good Morning", it is used the `hello_world` id that refers to the resource depending on the language set in the software settings.

```
/*
Localizable.strings (Finnish)
*/
"hello_world" = "Hyvää huomenta!";

/*
Localizable.strings (English)
*/
"hello_world" = "Good morning!";


let greetingsId = "hello_world"
let goodMorning = NSLocalizedString(greetingsId, comment: "")

print(goodMorning)
```

Figura 1. Internationalization process [Ynion 2020].

The *localization* process is more than purely translating all the strings from the software, it also requires adapting the strings for the right context and meaning. For example, the word "fits" can be applied in the context of something that can be embedded inside another thing, and it can also be used to say that something does not match with another thing. If we merely translate a sentence using this word without taking into account its meaning in the context, the result could be something completely different from the expected. Beyond the text, images and audio media can also be localized. There are some non verbal signs that can mean distinctive things depending on the country and they should also be considered.

3. Context and Motivation

The ultimate goal of this research is to improve the automation support for internationalization and localization testing.

Localization and internationalization testing ensure that the software meets the requirements of a local language, market and user's culture [Zhao et al. 2010]. It can be a very costly process in terms of time and money depending on the language scope and testing strategies. For instance, considering Google as an example: in 2011 the company went from 40 to 60 languages supported and more than 256 local languages versions (locales) [Google 2022]. Considering that Google needs to hiring a professional linguist to review the texts from its applications for all the 60 supported languages in order to confirm that the semantics, grammar, and syntax are adequate, it should be highly costly.

Localization testing is also a time consuming task because the tester needs to set the software's language for each supported locale and compare it against the source version, looking for layout inconsistencies, misspelled words, and crashes that can eventually occur for a specific locale. Furthermore, it is almost impossible for a human tester to be

fluent in all of the languages supported by the software under test, which makes the process of finding typos and grammar related issues more challenging. Given its time consuming nature, l10n testing is a very good candidate for test automation and we hypothesize that the current tooling support is far from ideal, leaving room for improvement.

Some of the most common failures found during localization and internationalization testing are: ellipsis, truncation, data-hour format, overlapping, and misspelled words. In Figure 2 (center), an example of the *overlapping* issue is displayed. This issue happens when the size of the translated strings are larger than the layout and the text overlaps other UI elements. The *truncation*, in turn, is an issue that occurs when the layout does not display all the translated string and it gets truncated. Figure2 (right) shows an example of the truncation issue: the translated string "MXN 750" does not fit in the layout and only the first word "MXN" is displayed to the user.

Figura 2. Localization and Internationalization most common failures.

4. Methodology

The proposed steps for the execution of this master thesis are displayed in Figure 3. The activities that will be develop in each step are detailed as follows.

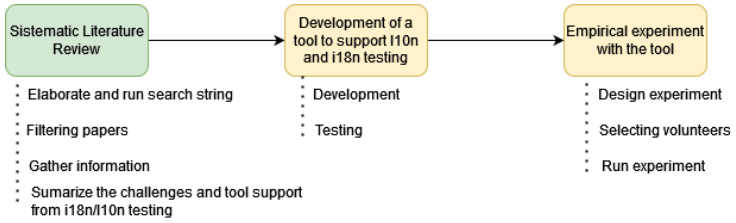

Figura 3. Proposed Master Project steps.

4.1. Systematic Literature Review

The first step, highlighted in green in Figure 3, is to perform a Systematic Literature Review (SLR) on i18n and l10n testing, which is the ongoing phase of this work. The objective of this SLR is to identify the state-of-the-art and practice of internationalization and localization testing. Guided by this motivation, the research question *RQ: What is the state-of-the-art in localization and internationalization testing?* was defined.

To answer this main question, the following specific objectives have been outlined for this SLR:

- To identify the main challenges faced by localization and internationalization testing.

- To detect which l10n and i18n testing strategies are the most used.
- To analyze the support tools for l10n and i18n testing.
- To establish if there are differences between the i18n and l10n testing techniques depending on the context in which they are used (e.g., games, mobile, web, desktop applications).
- To propose a new tool for l10n and i18n testing based on the SLR findings.

We have followed the Guidelines of Kitchenham [Kitchenham and Charters 2007] for performing the SLR. With the results of this Systematic Review we expect i) to discover what is the current state of practice in i18n and l10n testing; and ii) to produce a guideline for practitioners of localization and internationalization testing.

4.2. Development of a tool to support localization and internationalization testing

We plan to develop a tool to support i18n and l10n testing execution. With the results from the SLR we expect our tool to focus on the most promising strategies and on the actual needs of practitioners. Initially we will focus on the context of l10n testing for Android applications and our tool should be capable of automatically identifying the following types of l10n bugs: ellipsis, data-hour format, truncation, misspelled words, and overlapping.

For the first version of the tool it will receive as input an Android smartphone application. The tool will explore the app while taking screenshots and searching for i18n/l10n issues. At the end of the execution it should deliver a report of the potential issues identified: ellipsis, data-hour format, truncation, misspelled words, and overlapping. The generated report should also display a highlighted screenshot showing the location of the potential bugs. The proposed tool should be capable of supporting all locales that are enabled in the device.

4.3. Empirical evaluation

After developing the tool we aim to conduct an empirical evaluation in an industrial context following the guidelines defined by [Wohlin et al. 2012]. Our study will aim at investigating the effectiveness of our proposed tool into assisting i18n and l10n testers in their daily activities. The experiment will be carried out in a real industrial setting that performs i18n/l10n testing on Android devices.

5. Preliminary Results

At the time of this submission we are in the process of concluding the planned SLR (step 1 from Figure 3). We searched primary studies by querying the Scopus digital library, and by performing backward and forward snowballing cycles. We started with a set of 1073 papers and after applying our study protocol, we selected a final set of 24 primary studies. Currently we are extracting data from the selected primary studies to answer the research questions defined to our SLR.

6. Related work

From an *ad hoc* research made before starting the systematic literature review, two studies that related to this work were found:

The work of [Zhao et al. 2010] describes the localization testing models and which one is more appropriate for the resources and budgets available. They also describe the testing environment, phases and implementation process. However, the authors are more focused on the localization testing management point of view. Our work will be focused in producing a more complete essay by reporting the challenges, testing techniques and differences of approaches used in multiple platforms.

In [Ramler and Hoschek 2017b] the authors describe an approach that they applied for automated testing of the different localized variants of a large software product. Their contributions are in the empirical results of using an automated tool for the support of localization and internationalization testing for finding various bugs classified as critical and cosmetic issues. Furthermore, the authors provide practical insights on how to develop a scalable, industry-strength automation approach. It is important to highlight that the authors bring in up their work the attention for the lack of tool support and empirical studies available on the topic of i18n and l10n testing.

7. Expected Results

Carrying out effective internationalization and localization testing is crucial for providing a better experience for the users when using the software in their native language. After concluding the SLR we expect to deliver a guideline to be used by the localization and internationalization testing community. By developing an automated tool to support the localization testing process we expect that it can aid the testers in their daily activities. We hypothesize that the empirical evaluation of our proposed tool will show that it can effectively assist i18n and l10n testers in their daily activities.

Referências

Awwad, A. M. A. and Slany, W. (2016). Automated bidirectional languages localization testing for android apps with rich gui. *Mobile Information Systems*, 2016:13 pages.

Google (2022). Localization essencials by google. `https://www.udacity.com/course/localization-essentials--ud610`. [accessed 04-oct-2022].

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Ramler, R. and Hoschek, R. (2017a). How to test in sixteen languages? automation support for localization testing. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 542–543.

Ramler, R. and Hoschek, R. (2017b). Process and tool support for internacionalization and localization testing. *Product-Focused SW Process Improvement*, pages 385–393.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer, Germany.

Ynion, J. C. (2020). Using ai in automated ui localization testing of a mobile app. Master's thesis, Metropolia University of Applied Sciences.

Zhao, C., He, Z., and Zeng, W. (2010). Study on international software localization testing. In *2nd World Congress on Software Engineering*, volume 2, pages 257–260.

Štefan Molnár (2016). The software localization process. Bachelor thesis, Tomas Bata University in Zlín Faculty of Humanities.