



A general view of the localization of apps for mobile devices: status, challenges and trends. Formats and customary processes in the translation of iOS and Android apps

Eduard Simón
LocalVersion (Localization Manager)

ABSTRACT

Given the exponentially quickening inrush of mobile devices into our lives, predicting even the near-term evolution of this trend may be audacious, but is still essential. With this challenge in mind, this article tries to thresh out the scenarios that localization professionals may face when catching up with the latest trends on localization for mobile devices. Specifically, it details a typology of localization-industry clients and projects while exploring the processes and the most common formats involved. The author presents different categories of projects involving standard formats or complete Android or iOS apps.

Keywords: localization of apps for mobile devices, translation of iOS and Android apps

RESUM

El ritme amb què incorporem dispositius mòbils a les nostres vides s'accelera de manera exponencial. Per aquest motiu, qualsevol predicció de l'evolució d'aquesta tendència, fins i tot a curt termini, és arriscada, però tanmateix és necessària. Tenint present aquest repte, l'article intenta garbellar els escenaris amb què els professionals de la localització es poden trobar a l'hora de posar-se al dia amb les últimes tendències en localització per a dispositius mòbils. En concret, l'article descriu les tipologies de clients i de projectes de la indústria de la localització, alhora que n'explora els processos i els formats més habituals. L'autor presenta diverses categories de projectes que inclouen formats estàndards o aplicacions mòbils completes d'Android i iOS.

Paraules clau: localització d'aplicacions per a dispositius mòbils, traducció d'aplicacions d'iOS i d'Android

RESUMEN

El ritmo al que incorporamos dispositivos móviles a nuestras vidas se acelera de manera exponencial. Por este motivo, cualquier predicción sobre la evolución de esta tendencia, incluso a corto plazo, resulta arriesgada, si bien es igualmente necesaria. Con este reto presente, el artículo intenta cribar los escenarios con los que los profesionales de la localización se pueden encontrar a la hora de ponerse al día en las últimas tendencias en localización para dispositivos móviles. En concreto, el artículo describe las tipologías de clientes y de proyectos de la industria de la localización, a la vez que explora los procesos y los formatos más habituales. El autor presenta varias categorías de proyectos que incluyen formatos estándar o aplicaciones móviles completas de Android y de iOS.

Palabras clave: localización de aplicaciones para dispositivos móviles, traducción de aplicaciones de iOS y de Android

1. Introduction

More than likely you have used a multimedia telephone or digital tablet in the past few minutes: for Pokémon hunting, to reserve a table at your favorite restaurant, to check the weather forecast before your weekend trip, or to chat with friends in real time to organize your next outing ... routine actions that now form part of our day-to-day lives, with the convenience



of immediacy and ubiquity. As if that weren't enough, the new generations of wearables and the Internet of things push us into previously unexplored territory. The inrush of mobile devices into our lives is so exponentially aggressive that it proves audacious to predict its evolution even in the near term.

Immersed in this information explosion, the software localization movers and shakers are producing new and changing ingredients, formats, processes and commercial novelties. Every week, companies in the sector are confronted with the challenge of unplanned obsolescence.

In recent years, we specialized MLVs have observed a trending change in the typology of the localization projects assigned to us by software developers. When we speak about apps as opposed to desktop software, the traditional categories of language pairs by target market lose strength. Although "fr", "de", "ja", "es" and "it" maintain their leadership, it is no longer unusual to be faced with multilingual projects involving 25 languages. While some previously secondary markets are now gaining ground ("pt-BR", "ru", "ko", "zh"...), software manufacturers are trying their luck even in the traditionally closed and exotic markets of South East Asia or Eastern Europe. Although English maintains an indisputable leadership as source language, the explosion of app developers all over the map is causing the flourishing of exotic language pairs. At the same time, projects are shrinking when it comes to word counts, visual elements gain strength, and workflow accelerates at a frenetic tempo.

When freelancers and small to medium-sized LSPs venture into this market niche, they are confronted with the challenge of making a profit from projects that are inevitably small, global and urgent. The ability to manage efficiently across multiple time zones, handle complex encodings and letter types, and adapt to the administration and financial issues caused by agile operations in 30 different countries, is often a much greater hurdle than the technical and linguistic difficulties intrinsic to a localization project. The new management challenges alone merit rivers of virtual ink. In this article, we will try to thresh out the scenarios that localization professionals may face when catching up with the latest trends on localization for mobile devices.

2. Typology of clients and projects: formats and consequent processes

As in any sector of the translation market, a localization technician venturing into the world of mobile devices must take a commercial position in a market niche and define an ideal client. Simplifying the matter, we can establish three categories of potential clients: translation and/or localization agencies, small indie developers, and midsized or large developers. This choice will determine our workflow and the level of requirements for our capacity for adaptation and innovation. The translator can also segment by product category and opt to specialize in one of the multiple fields of software in the market: productivity tools, games, information security, banking, public entities ... As in so many other business sectors, the extent of specialization can be decisive in the success of a company.

Positioning will increase the probability of a vendor to be exposed to specific project types, but this is not necessarily cast in stone. In other words, although a localization agency will generally take on the more technical parts of a project and will provide the translators with pre-processed files for them to complete linguistic tasks, there may be times when translators are expected to assume total control over the technical part of the project (localization engineering). Likewise, a midsized software manufacturer will tend to send structures of localizable resources without processing them, or complete apps, but it will not be unusual to see it taking on a localization engineering role in certain scenarios. As a result, and independently of their origins, projects of app localization will tend to fall into one of the following categories:

Files pre-processed by the final client, and for which we will be faced only with challenges of translation in simple text formats or online platforms.



Translatable resources in standard formats that require a certain degree of processing or filtering (.strings, .xml, .xliff, .po, .json, etc.).

Complete apps for which we will have to assume partial or total control over the software project until its launch (.app, .apk, etc.).

What determines whether a client will opt for any given typology? Numerous elements come into play in this decision: the internal structure of the client, considerations of confidentiality, the available budget, preferences for control of the different phases of the process, deadlines, unawareness of the different paths available ...

In the lines to follow we will omit the projects of type 1, as they do not present technical challenges. Considerations generally more linguistic in nature are beyond the scope of applicability of this article.

2. 1 Translatable resources in standard formats

When we talk about translatable resources in standard formats, we are referring to any format based on simple text, tagging language or a format for data interchange that uses a systematic syntactic pattern of a “key/value” pair (or, in other words, an “identifier/translatable text” pair). Definitively, such files are the result of a previous process of internationalization in which the team programming the app in question has separated the text strings from the source code. In reality, we are talking about formats such that in the absence of other and better means, we could wind up translating the files involved with a simple text editor such as Notepad on having identified the translatable elements. In the following demo images, we have indicated the first translatable element with a yellow line and have underlined in blue the corresponding key or identifying string.

```
1  /*
2  @LANGUAGE en
3  @DIRECTORY Note
4  @FILE Edition.strings
5  */
6
7  /* Class = "UIButton"; normalTitle = "Sketch"; ObjectID = "0NA-e6-w7n"; */
8  "0NA-e6-w7n.normalTitle" = "Sketch";
9
10 /* Class = "UIButton"; normalTitle = "Math"; ObjectID = "0Y9-3T-tXZ"; */
11 "0Y9-3T-tXZ.normalTitle" = "Math";
12
13 /* Class = "UILabel"; text = "Set as doodle"; ObjectID = "0h9-Np-zri"; */
14 "0h9-Np-zri.text" = "Set as doodle";
15
16 /* Class = "UILabel"; text = "Export"; ObjectID = "5uI-H4-G25"; */
17 "5uI-H4-G25.text" = "Export";
18
19 /* Class = "UILabel"; text = "Clear all"; ObjectID = "6WA-dR-llf"; */
20 "6WA-dR-llf.text" = "Clear all";
```

Figure 1. Code fragment from an iOS **.strings** file viewed with a simple text editor



```
Android.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4   <string name="app_name">Circle Home</string>
5   <string name="dummy_button">Dummy Button</string>
6   <string name="dummy_content">DUMMY\nCONTENT</string>
7   <string name="retry_caps">RETRY</string>
8   <string name="scan">Scan</string>
9   <string name="continue_caps">CONTINUE</string>
10  <string name="connect">Connect</string>
11  <string name="check_status">Check Status</string>
12  <string name="congrats">Congratulations!</string>
13  <string name="pair">Pair</string>
14  <string name="ssid">SSID</string>
15  <string name="rssi">RSSI</string>
16  <string name="unknown">Unknown</string>
17  <string name="discover">Discover</string>
18  <string name="help">Help</string>
19  <string name="name">Name</string>
20  <string name="email">Email</string>
```

Figure 2. Code fragment from an Android .xml file viewed with a simple text editor. Color scheme activated for xml syntax

In practice, risks and inconveniences cause us to discard this possibility completely, and lead us to use professional localization tools that will, among other things, offer capabilities of interaction with translation memories, syntactic protection, customization of filters, update automation, and quality control systems. SDL Passolo and Alchemy CATALYST provide some of the most recognizable solutions in this field. Fortunately for the translator, nowadays these platforms incorporate systems for analysis (parsing) capable of automatically interpreting which are the translatable elements of the most common formats used in the internationalization of applications. More and more, the usual CAT tools used day to day by the translator also incorporate functions and filters that make it possible to process most standard formats.

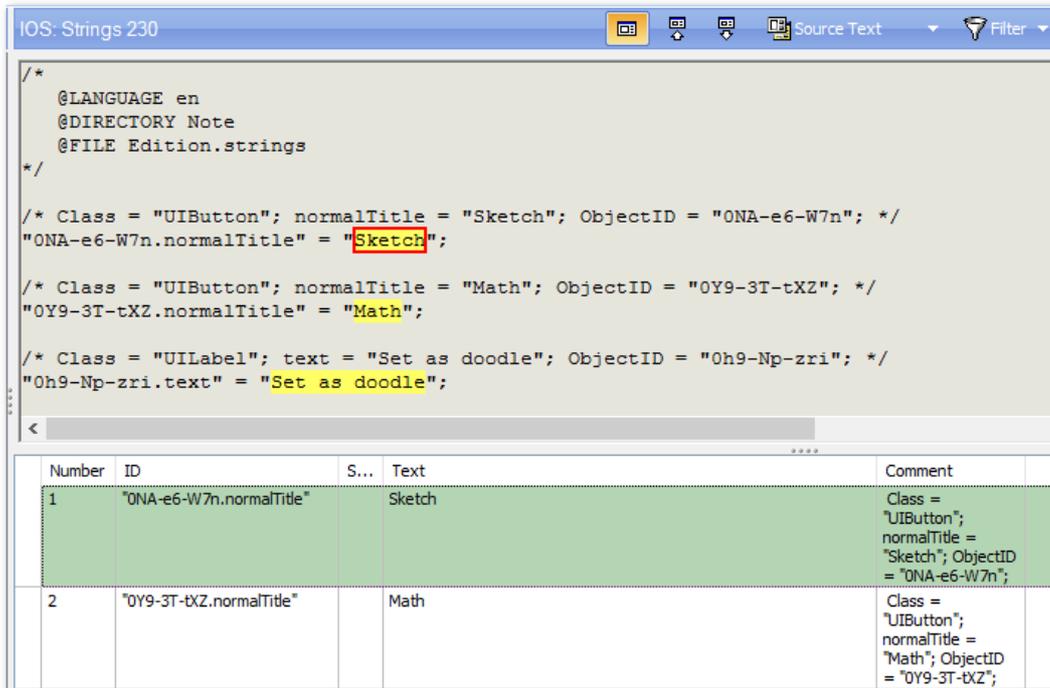
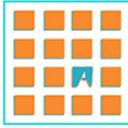


Figure 3. View of the SDL Passolo parser automatically interpreting the translatable strings in the **.strings** file from figure 1

But the most powerful functions in a professional localization suite are possibly to be found in its capacity to create personal filters to process unrecognized formats, its flexibility in processing and moving information in different formats and directions, and its delivery of visualization of data in languages with contextual information or bending of gender and number. Without the need for programming knowledge and by just learning a few simple techniques, an app localizer can have a most powerful ally available whenever new formats appear in the market or projects incorporating non-standard or unusual formats are encountered. Once the localizable resources have been processed with a tool with these characteristics, the possibilities of moving the relevant data to integrate it into the work process of any organization are unlimited: from tab-delimited CSV files to .TMX files, as well as all kinds of intermediate formats compatible with SDL Trados, MemoQ or OmegaT, to name a few.

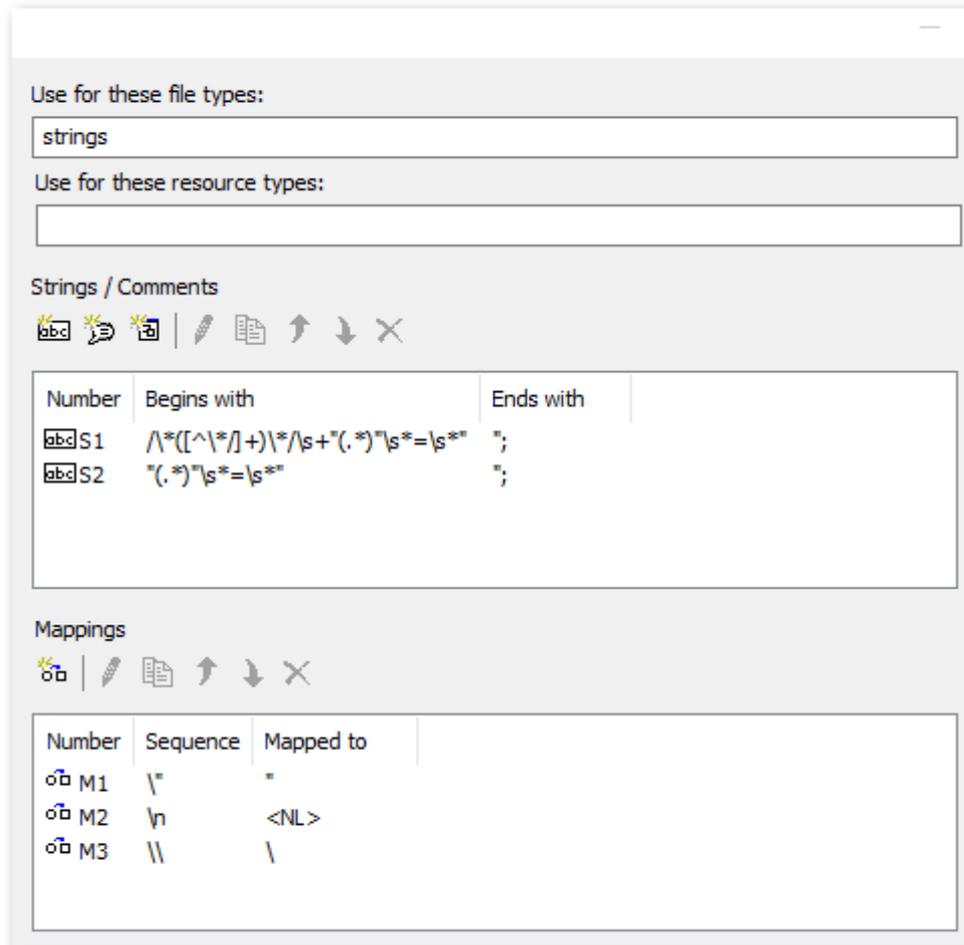
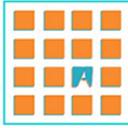


Figure 4. View of the function to create customized filters in SDL Passolo

The capability of automatically processing updates also proves extremely useful. If in any kind of software localization process it is key to have available an agile and precise system for updating versions, in mobile localization projects this aspect takes on vital importance. Let's not forget that we face projects with very short life expectancies, and consequently both urgent and susceptible to constant and rapid updates. In the development cycles of a product, it is not unusual for the localization team to receive new updates before having been able to complete the translation of the current version. The orderliness and precision of the localizer are, then, absolutely critical.

In their eternal search for ways to simplify and speed up localization cycles, the big players in the software industry (Apple, Google, Microsoft...) continually propose formulae to optimize localization processes. Obviously the practices established by companies of such global importance end up defining the standards used by smaller manufacturers. Therefore, it is essential to follow the evolution of their work systems closely, as they will determine what kinds of projects we can expect in the near future. Serving as an example is the 2014 introduction of the XLIFF export/import function in Apple products, beginning with XCode 6. Apple thus narrowed the boundary between programmer and localization engineer, with two important repercussions: converting the XLIFF standard into one well known by any localizer of iOS applications, and at the same time changing the process of localization, diverting part



of the technical work of localization engineering from the translation team to the development team.

```
en.xliff
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <xliff xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:xsi="http://
  www.w3.org/2001/XMLSchema-instance" version="1.2" xsi:schemaLocation="
  urn:oasis:names:tc:xliff:document:1.2 http://docs.oasis-open.org/xliff/v1.2/os/
  xliff-core-1.2-strict.xsd">
3   <file original="Locales/en.lproj/Localizable.strings" source-language="en"
  datatype="plaintext">
4     <header>
5       <tool tool-id="com.apple.dt.xcode" tool-name="Xcode" tool-version="7.3"
  build-num="7D175"/>
6     </header>
7     <body>
8       <trans-unit id="Accept">
9         <source>Accept</source>
10        <note>No comment provided by engineer.</note>
11      </trans-unit>
12      <trans-unit id="An unknown error occurred">
13        <source>An unknown error occurred</source>
14      </trans-unit>
15      <trans-unit id="An unknown error occurred, please check your internet
```

Figure 5. View of XLIFF format. Rather than replacing the source string, the translation would be appended in a new <target> tag in this case

And here we introduce an important concept in localization from the commercial perspective: processes evolve, work changes hands quickly, and new jobs appear but often disappear. We will see a great example when we discuss auto-layout in the next section. Continuous training and resilience are, then, key elements in the mobile localization sector.

For projects in this category, the client may have sent individual files, structures of standard files that follow the internationalization criteria of Apple (iOS) and Google (Android), or even work structures proprietary to specific developers. As in any software localization project, client expectations will include strict adherence to the structures of work directories, the integrity of file syntax, and that of character encoding, including subtle details such as the kind of line return used: Windows (CR+LF), Unix (LF) or Mac (CR).

A localization project for translatable resources in standard formats may (or may not) include a final quality assurance (QA) phase once the development team has integrated the translations into the product in question. Unfortunately, this phase is too often eliminated, either because of budget limitations or because of unawareness of the implications for the final quality of the product. Thus, it is not unusual to find, even in well-known apps, significant errors that could easily have been avoided with a few minimal checks preceding launch. In cases where a testing phase is to be properly completed, it may take place on the client side or in collaboration with the localization team. Beyond the pedagogical work that the localizer may perform, the flexibility to adapt to the specific requirements of each client will again be fundamental.



2.2 Complete apps

Some software companies seek full localization services on the part of professionals who will act in the manner of an internal department, taking responsibility for the entire process of translation and localization. In these cases, it is customary to receive complete mobile applications for adaptation in .APP (iOS) or .APK (Android) format, or else development structures with environments similar to those used by the programmers themselves. This kind of project requires great knowledge of the development tools involved, mainly Android Studio for Android apps and XCode for iOS apps, even if numerous third-party alternative solutions exist. It is advisable to have extensive localization experience before accepting whole projects; although the engineering team may participate from time to time, the client will expect the localizer to be able to operate autonomously in completing final products ready to be launched in new markets. In addition, for the most part, projects in this category are multilingual ones that will involve various professionals. Thus, the localization team will be responsible for preparing simplified translation kits.

The processing of complete applications requires tools for extraction and management of the content for the purpose of capturing all the translatable text and guaranteeing the homogeneity of the translation process. It is feasible to observe by hand the content of these apps, since after all they are simple file structures that are readily manipulated. Observe in the following table how, in reality, they are nothing more than executable containers of the types of file that we described in the preceding section, where translatable resources are separated from compiled binary code, available in turn in other directories. Also notice the use of ISO codes for languages to structure content by language.

Basic structure of an Android app	Basic structure of an iOS app
FILE.APK/ res/ values/ strings.xml values-es/ strings.xml values-fr/ strings.xml	FILE.APP/ Resources/ (optional folder) en.lproj/ Localizable.strings es.lproj/ Localizable.strings fr.lproj/ Localizable.strings

In practice, we will always use professional extraction tools for this kind of project. Directory structures tend to be rather more populated than in the previous basic example, and they use resource structures that are often divided into multiple levels. Besides being laborious and tedious, any effort at manual processing would make it too easy to miss translatable elements. Tools such as AppleGlott, iLocalize, Alchemy CATALYST or SDL Passolo are good allies to identify files of translatable resources automatically and to prepare translation kits. At the same time, this kind of tool will prove fundamental for controlling the continuity of the localization process and avoiding version mismatches.

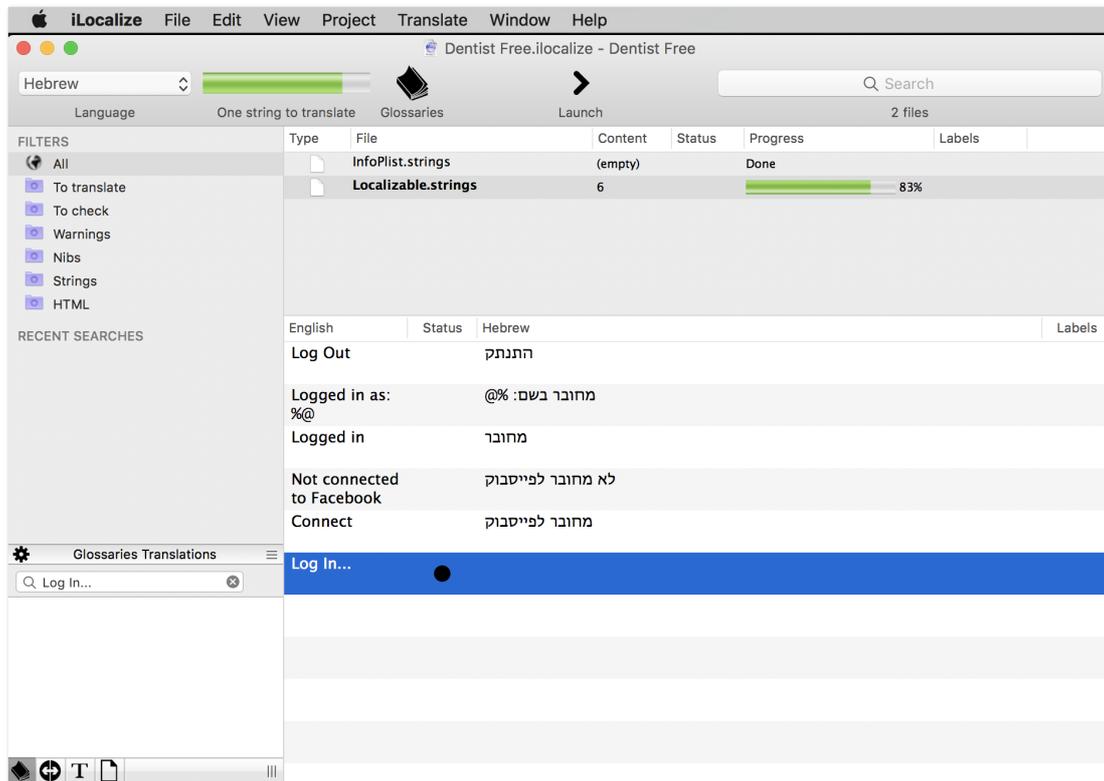


Figure 6. Processing an iPhone app with iLocalize

Until very recently, the translation phase was followed immediately and inevitably by a resizing phase. This would involve making adjustments to the user interface, and during this process the localization team would accommodate translations of various lengths, changing the sizes and alignments of the user interface. New advances in User Experience design, chiefly in techniques for responsive design and auto-layout and, to a lesser extent, the sophistication of available hardware, are eliminating this function, taking it off the game board at top speed. Every time strings of controlled lengths are being generated, the translator has to strive to shorten text when necessary, mainly during the quality control phases. Even though this requirement used to cause great agitation among translators in the early days, it is fair to say that now it is no longer the case. The evolution of hardware with mobile screens of 4.7 or 5.5 inches and the continuing improvements in typographic techniques have resulted in the translator's being able to return to writing in a relaxed manner, without the guillotine of dreaded truncations.

We have been touching lightly on the concept of responsive design. It is important to understand that this new set of technologies is greatly simplifying the work of the developer and the localizer. If in the early stages of localization for mobile devices it was common to encounter specific versions for various categories of telephones and tablets, the current trend is for a single app to be valid on different devices/screens. Moreover, the latest innovations with development engines such as Unity bring us to a multiplatform scenario starting from a single set of source code. The programmer generates just one set of code and the product is automatically deployed on numerous platforms and devices.

In the project typology that concerns us, the testing or LQA acquires an especially important dimension. The client will expect a correctly verified product, often without considering the methods used to be important. Roughly speaking, we will have three options



available: real physical devices (smartphones, tablets or wearables), emulators of mobile devices included in development SDKs, and online emulation platforms (Manymo or similar). Emulation systems present great advantages, primarily their reduced cost (or costlessness) compared to the cost of real devices, as well as their enormous configuration flexibility, in that they can be used to test various models and screens from a single location. The problems that we will detect and solve during this phase may be linguistic, functional or a combination of the two.

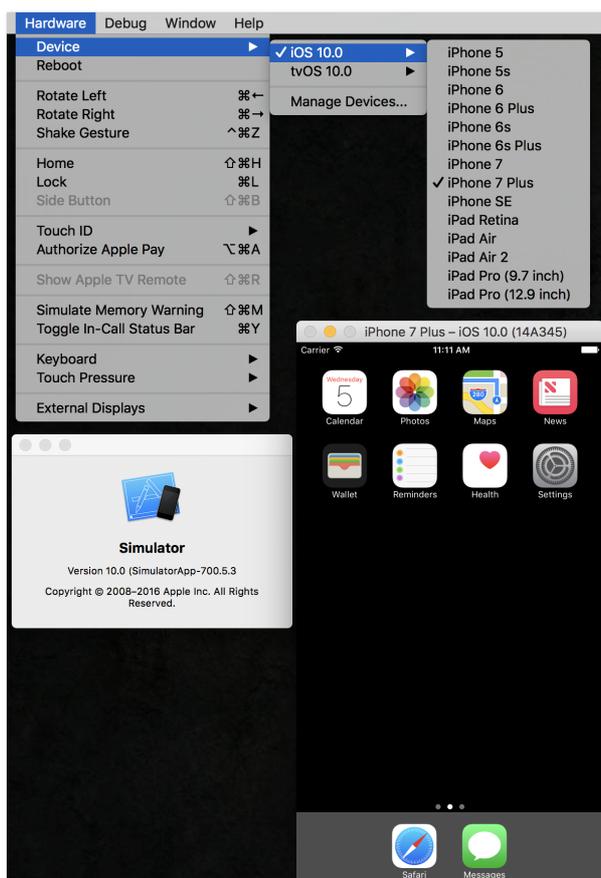


Figure 7. View of the selection of a device in iOS Simulator, included in Xcode

A few final thoughts about the proliferation of new generations of online platforms that enable localization entirely in the Cloud. Although it is indisputable that they offer significant advantages, especially regarding collaborative work and the management and organization of workflow, often they prove more convenient for the developer than for the translation professional, who may lose control of the project through not having customary work tools and methods available. Some considerations or risks of security, privacy and confidentiality are also arising; these are highly important elements in the world of software. Without a doubt, we will have to watch their evolution closely. Potentially they will have a predominant role in the near future.

3. Conclusion

In this general view of localization for mobile devices, we have tried to introduce some basic concepts for the localization professional who decides to venture into this market sector.



We would like to encourage you to continue exploring those concepts which we have been unable, by reason of obvious space limitations, to discuss in much depth.

Mobile localization is developing at a dizzying pace. The professional who decides to enter this market niche will do well to prioritize continuous training and to learn general localization techniques that can be adapted to various processes and situations, instead of memorizing mechanical work processes. The capacity for adaptation, flexibility and resilience will determine the survival of the various actors in the market. In a very demanding sector with volumes of shortened text, it will be interesting to explore linked ways of doing business meant to make the model profitable: complementary marketing materials, online store descriptions, web pages, advertising, technical documentation and, definitely, all the collateral materials that will accompany the applications when they are launched.

Useful resources

Localization Guide for Android Developers:

<https://developer.android.com/guide/topics/resources/localization.html>

Localization Guide for iOS Developers:

<https://developer.apple.com/library/ios/i18n>

Apple Computer Product User Guides:

<https://support.apple.com/manuals>

App Store Marketing Guidelines:

<https://developer.apple.com/app-store/marketing/guidelines/>

Apple Internationalization:

<https://developer.apple.com/internationalization/>

Android Open Source Project:

<http://source.android.com>

Android OS strings (English):

<https://android.googlesource.com/platform/packages/apps/Settings/+master/res/values/strings.xml>

Android OS strings (Spanish):

<https://android.googlesource.com/platform/packages/apps/Settings/+master/res/values-es/strings.xml>

Microsoft Language Portal

<https://www.microsoft.com/Language/>