

Proceso de testing en localización: análisis y propuesta de un modelo de codificación de errores, de informe de errores (*bug report*) y de *test plan*

Beatriz Bailén Ruíz¹

Universidad Internacional Menéndez Pelayo

Plaza Puerta de Jerez, 4 · 41001 · Spain

RESUMEN

El debate en torno al concepto de error está presente desde los comienzos de los Estudios de Traducción. Debido al avance tecnológico en un mundo globalizado como el de hoy se ha producido un aumento en la prestación de servicios de localización; un proceso que abarca un mayor número de tareas de distinta índole aparte de la traducción de texto. Los objetivos de esta investigación son describir y analizar conceptos a modo de recopilación teórica, determinar los parámetros necesarios para proceder al testeo de productos localizados y estandarizar y sistematizar el proceso de testeo. Se plantea también la necesidad de analizar los distintos tipos de errores que pueden darse en un proceso de localización, así como realizar una taxonomía de estos, proponiendo una codificación y creando un modelo de informe aplicable en la fase de revisión del proceso de localización junto con unas instrucciones de uso para el localizador, para así unificar el proceso de revisión y testeo, con el consiguiente aumento de la calidad del producto final.

Palabras clave: testeo, taxonomía de errores, codificación de errores, testeo lingüístico, testeo funcional, informe de errores.

ABSTRACT

The concept of translation error has been under discussion in Translation Studies since its beginnings. The technological advance in a globalised world like this implies an increase in the provision of localisation services; a process which includes a greater number of tasks other than translating text. The objectives of this research are to describe and analyse theoretical concepts as a theoretical review; to determine the necessary parameters to carry out a testing task of localised products and to standardise and systematise the testing process. There is also a need to analyse the types of errors being made in a localisation process, as well as to conduct a taxonomy of these errors and propose an error encoding. In addition, a bug report model applicable in the revision phase of the localisation process is created together with a set of instructions for the localisers with the aim of unifying the process of revision and testing, increasing the quality of the final product.

Keywords: testing, taxonomy of errors, error encoding, linguistic testing, functional testing, bug report

1. Introducción

Nos encontramos en un mundo globalizado en el que el conocimiento y la cultura son compartidos y transmitidos entre diversas naciones y donde la traducción ha sido siempre ese puente que une culturas y transmite conocimientos.

Unido al concepto de globalización está el de internacionalización, que, aplicado al mundo de la localización, se define como la preparación de productos desde su creación para que puedan ser adaptados y transmitidos a otras culturas con mayor facilidad. La localización se enmarca entre estos conceptos, puesto que consiste en adaptar un producto (generalmente de componente tecnológico) a un mercado local concreto. Este concepto va más allá de la traducción, ya que esta adaptación no es solo lingüística, sino cultural. El auge de la internacionalización

¹ Corresponding author · Email: beatriz_bailen@hotmail.com

y la globalización de mercados tiene como resultado un aumento de la prestación de servicios de localización, y que, por lo tanto, suscita el interés de muchos profesionales del mundo de la traducción.

Es sabido que la revisión en traducción es crucial para afirmar la calidad de un texto y el testeo es igualmente importante para un producto localizado. Este trabajo de investigación se concreta en torno a una fase del proceso de localización que consideramos de vital importancia: el testeo o testing de un producto localizado; es decir, la revisión de un producto en su contexto de funcionamiento tras aplicar los diferentes cambios en un proceso de localización.

La justificación e importancia de este trabajo se encuentra en la relevancia que tiene el testeo en localización y la necesidad investigadora que suscita, puesto que se trata de una práctica en auge y supone un servicio de valor añadido para el traductor que se ha formado en localización. Por otro lado, afirmamos que una estandarización y sistematización de un proceso como este supondría una mejora en la calidad del servicio; y esta investigación se estructura a partir de esta premisa.

En este artículo se va a analizar el estado de la cuestión sobre el testeo en localización, el error en localización mediante una taxonomía de errores, los tipos de testeo, el uso de informes de error o bug reports en localización y la importancia de las instrucciones o test plans para la realización de este proceso.

1.1. Objetivos

Los objetivos generales de este trabajo parten del análisis de la práctica de revisión y testeo en localización, así como de la necesidad de estandarizar este proceso. Los objetivos generales de este trabajo son:

- describir y analizar conceptos teóricos a modo de unificación de conceptos recopilando una guía teórica en torno al proceso de testeo en localización;
- determinar los parámetros necesarios para proceder al testeo de productos localizados.

Los objetivos específicos giran en torno a la aplicación práctica de las conclusiones extraídas del análisis de la información recopilada para crear herramientas prácticas. Estos son:

- proponer una taxonomía y un modelo de codificación de errores para facilitar su identificación en un informe de errores;
- proponer un modelo de informe de errores o bug report para la fase de testeo en localización a modo de ejemplo práctico y
- analizar el uso de dicho modelo y proponer una plantilla de instrucciones a modo de encargo para el revisor.

2. Antecedentes y estado de la cuestión

2.1. Antecedentes

En cuanto a la estructuración teórica en torno a la práctica del testeo en localización, es importante mencionar los conceptos relacionados con el testeo de *software* que Luo (2001) plantea. Esta autora determina que la fase de testeo ocupa un 40 o 50 % de los esfuerzos en desarrollo de *software*; además, afirma que esta actividad sigue siendo la forma más efectiva de asegurar la calidad de una herramienta localizada y añade que será cada vez más importante en el futuro.

En su trabajo, se centra en las claves para la investigación del testeo de *software* y las distintas teorías investigadoras de otros autores hasta la fecha. Consideramos importante incluir la definición que incluye citando a E. F. Miller (1981):

The general aim of testing is to affirm the quality of software systems by systematically exercising the software in carefully controlled circumstances (Luo, 2001, p. 1).²

En cuanto a la teorización que Luo (2001, p. 1) realiza en torno al proceso de testeo, presenta un análisis de lo que llama *The testing Spectrum*: el testeo se puede aplicar en todas las fases de desarrollo de un producto y puede realizarse desde unidades mínimas hasta del programa en su totalidad. Distingue, por lo tanto, cuatro tipos de testeo: a) *unit testing*, que se realiza a las unidades mínimas objeto de testeo; b) *integration testing*, con dos o más unidades integradas en un sistema más amplio; c) *system testing*, en el que se revisa la totalidad del programa; y d) *acceptance testing*, cuando el producto se envía a los consumidores finales, con el objetivo de determinar si el producto funciona en el mercado y buscar errores.

Respecto a las técnicas de testeo encontramos por un lado el *análisis estático*, en el que se realizan tareas de revisión antes de ejecutar el programa; y el *análisis dinámico*, que recoge pruebas realizadas con el producto ejecutado. Por otro lado, distingue el *testeo funcional*, que se realiza para evaluar la funcionalidad y el comportamiento externo del producto; y el *testeo estructural*, que enfoca su actividad en detectar errores en el funcionamiento de la estructura interna del producto y en relación con el sistema operativo en el que se ejecuta.

Como hemos podido ver, el testeo puede realizarse desde distintos enfoques. El proceso de revisión puede tener, en consecuencia, distintos objetivos. La elección de una o más técnicas de testeo para cada proyecto serán diferentes según el objetivo requerido para cada proceso de revisión y testeo.

Por otro lado, Shanthi (2006) hace hincapié también en la importancia de la internacionalización de los productos para facilitar la labor de localización. Además, aporta un modelo de informe de verificación para el proceso de testeo bastante completo a modo de *checklist*. La autora aporta una tipología de revisiones en localización parecida a la de Luo (2001, p. 8) en cuanto a los conceptos de *testeo funcional* y *estructural*, aunque aporta nuevos conceptos como *user interface validation*, que consiste en revisar errores estéticos o de apariencia como el texto truncado y solapado o problemas de gráficos; *usability testing*, que evalúa el uso dinámico del producto; *installation testing*, que mantiene como opcional y consiste en verificar que los mensajes que aparecen durante

² «El objetivo general de testeo es verificar la calidad de los programas informáticos mediante la ejecución sistemática de los programas en situaciones cuidadosamente controladas».

la instalación de un producto están correctamente traducidos al idioma meta; y *localization testing*, que define como una verificación del contenido lingüístico que se centra principalmente en la idoneidad de la traducción de elementos como: el contexto de la interfaz de usuario, los archivos de ayuda, los mensajes de error, cuadros de diálogo, tutoriales o archivos *readme* y los documentos anexos como el manual de usuario y las instrucciones de instalación e información de la versión del producto.

La asociación LISA (*Localization Industry Standards Association*) define la localización como «el proceso de adaptación y fabricación de un producto para que presente el aspecto y el estilo de un artículo fabricado en un país en concreto» (LISA, 2003, p. 3) y argumenta su multidisciplinaridad en tanto que abarca no solo un proceso de transferencia lingüística, sino que también «impregna todos los aspectos de diseño, desarrollo, pruebas, mercadotecnia y asistencia de los productos. En otras palabras, unifica, y también trasciende, los servicios y disciplinas individuales como la ingeniería, traducción y acabado» (LISA, 2003, p. 3). Con el paso de los años, la localización ha ido aumentando su presencia en los servicios que prestan las agencias y las empresas de traducción en todo el mundo; y es que la localización «es la pieza de un rompecabezas comercial mundial que permite que las empresas mantengan negocios en mercados ajenos al suyo original» (LISA, 2003, p. 3).

La localización es, por lo tanto, la adaptación de un producto a un mercado local concreto, en el que será comercializado. Se trata de un proceso en el que intervienen distintas fases, según la naturaleza del proyecto localizado, entre ellas la de revisión o testeo. La globalización de los mercados y la internacionalización entendida como la preparación de productos para que sean adaptados lingüística y culturalmente a mercados internacionales suponen para la industria de la localización y la traducción un verdadero empuje. Globalización, internacionalización, localización y traducción forman las siglas GILT, utilizadas para explicar este fenómeno.

2.2. Estado de la cuestión

La cuestión objeto de estudio se encuentra en cierto modo desplazada desde un punto de vista de pertenencia a los Estudios de Traducción. Bien es cierto que el testeo es una fase que forma parte del proceso de localización; y la investigación dentro del mundo de la localización está comenzando a ganar terreno en la actualidad, donde encontramos que el testeo se enmarca dentro de artículos, guías o modelos de diferentes organizaciones sobre la calidad de las traducciones. Sin embargo, el material sobre este tema es normalmente de naturaleza práctica y aplicada, y su tradición en la investigación comienza en un primer lugar con el *testing* de *software* con fines de desarrollo en este campo, y no en el avance de la localización propiamente dicha dentro de los Estudios de Traducción. En este aspecto es necesario que este trabajo defina los conceptos que engloban el proceso de testeo aplicado a la localización.

En general se encuentran referencias a este proceso dentro de guías de gestión de proyectos y localización; sin embargo, existe poco material que estudie este proceso desde la perspectiva de la investigación en Traductología; es decir, es un tema sobre el que se ha teorizado poco, y ante el cual es necesario entrelazar distintos enfoques más consolidados en los Estudios de Traducción como puede ser el concepto de error y el de revisión de traducciones. Por lo expuesto, el estado actual de la cuestión se enmarca en torno a una revisión de la bibliografía actual más importante sobre el tema específico. Se basa, pues, en estudios aplicados a la práctica de localización de *software* y videojuegos, de los cuales se extraen las ideas que conciernen a nuestra cuestión, abordándolas desde una perspectiva descriptiva y analítica.

Consideramos importante la teorización sobre el proceso de testeo que Luo realiza en 2001, ya que aporta conceptos importantes en cuanto a las distintas tipologías de testeo y perspectivas, que recogimos en el apartado anterior. En su trabajo explica también el estado de la investigación de este campo, sobre el cual afirma que en la investigación sobre el testeo de *software* se ha llegado a resultados bastante sólidos. Sin embargo, añade que «testing remains an awkward, time-consuming, cost-ineffective chunk of work that is always not very satisfying in most industry practices» (Luo, 2001, p. 14).³ Además, habla de la existencia de lo que llama *testing gap*, ya que solo una parte de los resultados de la investigación en este campo se han utilizado con éxito en la práctica empresarial para mejorar el proceso de testeo; y concluye:

It is always the case that testing ends up being a must-end activity because the project runs out of budget and is beyond deadline. This inconsistency of testing research and practice has been called the testing gap (Luo, 2001, p. 14).⁴

El proceso de testeo de videojuegos se encuentra explicado con detalle en Chandler y Deming (2012). Este trabajo y el de Esselink (2000) suponen un conjunto de guías de localización imprescindibles; sin embargo, Chandler y Deming lo estudian de un modo más exhaustivo y justifican el uso de informes de error tanto en el testeo funcional como en el lingüístico. Lo que caracteriza al trabajo de Chandler y Deming, aparte de analizar con mayor detalle este procedimiento, es la posibilidad de abstracción de la práctica y poder extraer de sus líneas generalizaciones teóricas en cuanto a esta práctica. Los parámetros generales que establecen son los siguientes:

- Separación del proceso de testeo funcional y del testeo lingüístico, aunque en el funcional pueda informarse de errores lingüísticos y en el lingüístico de errores funcionales.
- La importancia de definir las pautas para cada proceso de testeo, quiénes son los responsables y dar las instrucciones (test plan) necesarias con el mayor número de detalle para facilitar dicho proceso.

El testeo debe ser metódico y los errores deben registrarse de forma unificada, informando de cualquier tipo de error utilizando un informe de errores (bug report) para que los errores puedan identificarse y solucionarse de la forma más eficiente.

Como conclusión a este apartado de contextualización teórica podemos afirmar que, sin lugar a dudas, es posible ampliar la investigación en técnicas de testeo, ahondar en cuestiones teóricas y trabajar en cuestiones prácticas, tales como las que se proponen en este trabajo, y muchas otras relacionadas con aplicaciones informáticas de *software* de testeo y de informes de error en localización.

3. Metodología

En primer lugar, la metodología comienza por establecer un marco teórico sobre el proceso de testeo que nos permita contextualizar; en segundo lugar, elaborar una taxonomía de errores y un análisis de las herramientas utilizadas en la fase de testeo, en este caso el informe de errores o *bug report* que se propone como modelo.

³ «El testeo sigue siendo un trabajo tedioso y poco rentable que no siempre resulta satisfactorio en la mayoría de las prácticas profesionales».

⁴ «Siempre ocurre que el testeo termina siendo una obligación porque el proyecto se queda sin presupuesto y sobrepasa la fecha de entrega. Esta incoherencia entre la investigación sobre el testeo y la práctica se denomina ‘brecha del testeo’».

La primera fase de contextualización teórica será descriptiva y analítica. Tanto en el apartado de marco teórico como en el de estado de la cuestión se estudia el estado de este campo disciplinario, describiendo a modo de fase reconstructiva la realidad del proceso de definición y clasificación de errores en la fase de revisión o testeo de un producto localizado.

A continuación, la segunda fase, de aplicación práctica, parte de una perspectiva empírico-analítica de diferentes modelos de informes de errores o *bug reports* que se recogen en distintas fuentes documentales con el fin de proponer una taxonomía de errores en localización junto a un modelo propio basado en la información analizada y los datos expuestos. Por último, dentro de esta fase, y como complemento al modelo de informe de errores, se adjunta una plantilla de instrucciones de uso (*test plan*) del modelo de *bug report* propuesto.

El análisis y la discusión teórica de la primera fase parten desde el marco teórico de modo general y en el apartado de análisis se extraen las conclusiones relacionadas con la aplicación práctica, en el que se realiza una breve discusión teórica. De este punto se extraerá la metodología de revisión propuesta en este trabajo, la taxonomía y codificación de errores y el modelo de informe de errores propuesto; elementos que constituyen la aplicación práctica de este trabajo.

4. Taxonomía de errores en localización

4.1. Estéticos

Son los errores relacionados con la apariencia y la visualización del producto localizado. Estos errores se suelen encontrar en la interfaz de usuario. La apariencia del producto localizado debe corresponderse con la del original, en cuanto a la disposición de los elementos como p. ej. los cuadros de diálogo, menús, y mensajes del sistema, etc. Aquí nombramos algunos errores de este tipo a partir de la checklist que Esselink (2000, p. 151) nos propone en su guía de localización:

Errores estéticos
Falta un elemento en la interfaz
No se ha localizado un elemento en la interfaz y aparece en el idioma origen
Faltan opciones en un menú
El tamaño de un elemento está sin reajustar
Texto truncado
Texto solapado por otros elementos
Los caracteres especiales no se muestran correctamente
Un elemento se muestra mal alineado en la interfaz
Una imagen, botón o icono no se visualiza correctamente
Imágenes o iconos sin localizar
El texto de los enlaces no está traducido
Un elemento de audio o video no están localizados
Formatos de fecha, hora, divisa o calendario sin localizar
Contenido no adecuado para la cultura meta (elementos estéticos)
Error de apariencia: geometría, fondos, estética
Error en el tipo de fuente

Tabla 1. Errores estéticos.

4.2. Funcionales

Se trata de errores que afectan a la funcionalidad del producto localizado. Normalmente, estos errores se han introducido durante el proceso de localización. Este tipo de testeo también se realiza en la versión original antes de introducirse en el mercado y antes de localizarse. El testeo funcional del producto localizado es igual que el del producto original, pero se diferencian en que la aplicación localizada se ejecuta y se pone en funcionamiento en un sistema operativo que ya está localizado y adaptado a la región o locale meta para asegurarse de que no hay incompatibilidades entre el producto localizado y el sistema operativo en el que se va a usar.

(1) Localización de *software*

Errores funcionales en localización de <i>software</i>
No se muestra alguna opción de menú
Un elemento no funciona correctamente
Se ha asignado a una tecla de atajo una función que no tenía en el original
No se ha asignado una tecla de atajo a una función que sí tenía el original
Se ha asignado el mismo acelerador a más de una función dentro del programa
Tecla de atajo duplicada
Un acelerador no funciona
Un dispositivo o periférico de entrada no funciona en el programa
No se puede visualizar un documento adjunto
No se pueden introducir caracteres especiales
La aplicación no funciona correctamente en el sistema operativo
Un elemento que contiene una variable no está correctamente concatenado
La aplicación no abre un enlace correctamente
Los archivos de ayuda no se muestran correctamente

Tabla 2. Errores funcionales en localización de *software*.

(2) Localización de páginas web

Errores funcionales en localización de páginas web
Un enlace interno no funciona
Un enlace externo no funciona
Un botón no funciona correctamente
Un formulario de comunicación no funciona
Contenido audiovisual que no se visualiza
Un elemento no funciona correctamente en el cambio de navegador
El texto de los enlaces no está traducido
No se han utilizado rutas relativas para imágenes e hipervínculos
Uso incorrecto de CSS
Orden incorrecto de etiquetas
Faltan etiquetas
Sobran etiquetas que dan lugar a error
Uso de etiquetas JavaScript sobre lenguaje HTML
Codificación del texto no acorde
Atributos de HTML no válidos

Tabla 3. Errores funcionales en localización de páginas web.

(3) Localización de videojuegos

Errores funcionales en localización de videojuegos
Los botones del mando y los comandos en pantalla no se corresponden
Los dispositivos y periféricos de entrada no funcionan
Las variables incluidas en el texto no están correctamente localizadas

Incompatibilidad con el componente multijugador
Incompatibilidad con la plataforma online
El enlace a la documentación no funciona
Una opción o función no se muestra
Una opción o función no funciona correctamente
El audio no se muestra correctamente
Los subtítulos no se muestran
Los subtítulos no están sincronizados

Tabla 4. Errores funcionales en localización de videojuegos.

4.3. Lingüísticos

Son errores relacionados con la traducción de los elementos textuales y culturales. Consiste en verificar que se han traducido todos los elementos correctamente y que dicha traducción mantiene una calidad lingüística conforme a las normas gramaticales y de estilo; así como la adecuación al público receptor. Recogemos una serie de errores lingüísticos o de traducción, a partir de criterios propios y basándonos en la clasificación que Arevalillo realiza a partir del análisis y evaluación de traducciones (2012, pp. 15-19):

Errores lingüísticos
Error ortográfico
Error ortotipográfico
Error gramatical
Error de sintaxis (gramatical)
Error de tiempo verbal (gramatical)
Error de omisión
Error de adición/sobretaducción
Incoherencia terminológica
Traducción no funcional
Error de adaptación cultural
Error de sentido
Error de calco literal
Préstamo innecesario
Redacción incoherente
Error mecanográfico
Error de estilo

Tabla 5. Errores lingüísticos.

5. Propuesta de modelo de codificación de errores

La idea de clasificación de errores mediante un código viene dada por la voluntad de optimizar la tarea de registro y conseguir una mayor eficiencia en cuanto a comunicación eficaz y ahorro de tiempo. La codificación de errores se basa en criterios de economía lingüística y abreviación para estandarizar la tipología de errores más comunes en localización y ahorrar tiempo y espacio en la nomenclatura de errores en el informe de registro de errores (*bug report*).

La codificación se realiza atendiendo a la terminología en español, mediante símbolos en algunos casos y sobre todo abreviaciones de las primeras letras de cada palabra o término. Sin embargo, consideramos útil añadir otra propuesta equivalente en inglés, ya que este criterio es aplicable a todos los idiomas, puesto que parte de la

abreviación y simbología se crea a partir de criterios subjetivos; y una empresa de localización podría crear una simbología como esta estandarizada y aplicable a cualquier idioma, y que en el espacio correspondiente del registro se especifique el idioma del que se trata. Por eso proponemos el inglés como lengua para esta codificación estandarizada. Puesto que se trata una clasificación subjetiva, cualquier persona puede seguir estos criterios o establecer los suyos propios; la importancia radica en el objetivo de esta codificación y en el proceso de abstracción de los tipos de error más comunes basándonos en la clasificación expuesta en el punto anterior.

A partir de estos ejemplos se propone un método de clasificación, que, a través de un proceso mnemotécnico, puede agilizar el proceso de registro de errores, y que ayudaría a un equipo de testeo, desde el responsable de realizar el *testing* hasta el encargado de solucionar los errores a partir de la interpretación de estos en el *bug report*.

A partir de un proceso de abstracción de los tipos de error más comunes y la terminología de errores lingüísticos se propone para la clasificación de errores la siguiente codificación:

English (EN)	Code	Español (ES)	Código
Missed	?	Faltar	?
Not localized/translated	NL	No localizado/traducido	NL
Truncated text	TT	Texto truncado	TT
Duplicated	x2	Duplicado	x2
Not resized	NR	Sin reajustar	SR
Not displayed	ND	No se muestra/visualiza	NM
Does not work	NW	No funciona	NF
Does not run	DNR	No se ejecuta	NE
Not necessary/required	X	Sobrar	X
Incorrect/invalid	*	Incorrecto/no válido	*
Does not match	NM	No se corresponde	NC
Incompatibility	INC	Incompatibilidad	INC
Not synchronized	NS	No sincronizado	NS
Misspelling	MSS	Error ortográfico	ORT
Punctuation error	PE	Error ortotipográfico	ORTT
Grammatical error	GR	Error gramatical	GR
Syntax error	STX	Error de sintaxis	STX
Verb tense error	VB	Error de tiempo verbal	VB
Omission	OM	Omisión	OM
Addition	ADD	Adición	AD
Terminological inexactitude	TIN	Incoherencia terminológica	INT
Not functional translation	NFT	Traducción no funcional	TNF
Cultural adaptation error	CUL	Error de adaptación cultural	CUL
Cosmetic error	CE	Error estético/de apariencia	EE
Nonsense/misinterpretation	NSE	Error de sentido	SE
Literal translation	LT	Calco literal	CL
False calque	FC	Préstamo innecesario	PI
Inconsistent wording	IW	Redacción incoherente	RIN
Typographical error	TE	Error mecanográfico	EM
Stylistic error	STY	Error de estilo	EST

Tabla 6. Codificación de errores.

A partir de esta codificación proponemos algunos ejemplos de cómo se clasificarían errores como los expuestos en el apartado 4:

Código	Significado
TT Menú principal	Texto truncado en el menú principal
Cuadro de diálogo SR	Cuadro de diálogo sin reajustar el tamaño
Fecha y hora SL	Formatos de fecha y hora sin localizar
Imagen NM	No se muestra/visualiza la imagen
Acelerador Ctrl+P NF	El acelerador Ctrl+P no funciona
EtiquetaX	La etiqueta no es necesaria
GR Submenú	Error gramatical en el submenú
Orden etiquetas*	Orden de etiquetas incorrecto
INT Mensaje bienvenida	Incoherencia terminológica en el mensaje de bienvenida
TNF Instrucciones	Traducción no funcional en las instrucciones de uso

Tabla 7. Ejemplo de codificación de errores y su significado.

Llevando a cabo una nomenclatura como esta en un registro conseguimos:

- unificar el proceso de nomenclatura de errores;
- clasificar los errores por tipologías;
- ahorrar espacio y tiempo;
- posibilitar aislar con la herramienta de búsqueda y selección los errores de un mismo tipo;
- evitar errores de terminología;
- facilitar su ampliación añadiendo la codificación del idioma (por ejemplo: «ES_GR_Submenú») o añadiendo otro tipo de codificación a otros campos del informe de registro de errores.

Aunque bien es cierto que una codificación excesiva en la nomenclatura puede causar cierta ambigüedad en la correcta identificación del error, una codificación controlada combinada con palabras del lenguaje natural y términos del proceso de localización, como hemos utilizado en nuestra propuesta, es quizá la solución más efectiva ante una codificación excesiva.

6. El informe de errores (*bug report*) en el proceso de testeo

6.1. Breve discusión teórica

Dependiendo de la bibliografía consultada, según los diferentes tipos de testeo que contemplan, la limitación entre los errores que se deben evaluar varía. Algunos errores, por ejemplo, como el texto sin traducir, el texto truncado o el funcionamiento incorrecto de las teclas de atajo son considerados por Esselink (2000, p. 150) y por Chandler y Deming (2012, p. 237) como errores que se deben registrar en el testeo lingüístico. Algunos errores se repiten en la clasificación de Esselink como es el caso del texto truncado y la visualización incorrecta de caracteres especiales, que se contemplan tanto en el testeo lingüístico como en el estético (2000, p. 151).

En este sentido, este trabajo ofrece una clasificación bajo criterios propios, estableciendo límites entre los errores considerados estéticos, funcionales y lingüísticos; considerando estéticos aquellos errores que afectan a la visualización correcta de la interfaz, atendiendo a elementos visuales y la correcta disposición del texto, por ejemplo, el texto truncado y el texto sin traducir son errores estéticos, pues no significa que haya un error de

traducción ni que afecte a la funcionalidad, simplemente afectan la correcta visualización de la interfaz. Los errores funcionales son, por lo tanto, aquellos que afectan al correcto funcionamiento del producto; y los lingüísticos, los estrictamente relacionados con la tarea de traducción, atendiendo a criterios de funcionalidad en la cultura meta, la corrección lingüística y coherencia terminológica.

En cuanto a quién debe encargarse de cada tipo de testeo, Chandler y Deming (2012) determinan que quien debería realizar el testeo funcional del producto localizado es el equipo que se ha encargado de hacer el testeo funcional del producto original (2012, p. 219), y el desarrollador, el encargado de corregir los errores registrados (2012, p. 230), mientras que el testeo lingüístico lo realizan hablantes nativos que forman parte del grupo de traductores que han realizado la traducción del original, que revisarán el trabajo realizado (2012, p. 237). Por otro lado, Esselink considera que el testeo funcional debe realizarlo un equipo especializado en testing «*testing engineers*» (2000, p. 152) y el lingüístico debería realizarse por parte de un traductor especializado con conocimientos lingüísticos y de localización «*software translator*» (2000, p. 150). En este sentido, estamos de acuerdo en la importancia de que los encargados de realizar el testeo lingüístico tengan formación en traducción y localización.

Sin embargo, Chandler y Deming contemplan que en el testeo lingüístico se puedan registrar errores funcionales y en el testeo funcional errores lingüísticos: «Some critical functionality bugs may be reported by the linguistic testers [...] These functionality bugs need to be reported to the developer in detail right away so they can be fixed» (2012, p. 223).⁵ Por esta misma razón, consideramos que quien realiza el testeo de la herramienta localizada puede ser capaz de identificar y registrar tanto errores lingüísticos y estéticos como funcionales: «This is especially helpful if the linguistic testers report any functionality bugs because this allows the developer to quickly identify which version of the game has the bug» (2012, p. 218)⁶; por lo tanto, los diferentes tipos de testeo pueden hacerse por la misma persona, y llevar a cabo un correcto registro para su diferenciación es crucial.

6.2. Propuesta de modelo de informe de errores (*bug report*)

6.2.1. Funciones y objetivos

A lo largo de este trabajo hemos construido una argumentación en torno al uso de informes de error en el proceso de testeo. El objetivo de un *bug report* es dar la información necesaria para poder identificar, localizar y corregir los errores encontrados en un testeo. Cuanto más sistematizado sea este proceso, se conseguirá mayor eficacia en la tarea. Una estandarización en la nomenclatura y la creación de un *bug report* uniforme y generalizado serían ejemplos de sistematización del proceso de testeo.

La función de un *bug report* es actuar a modo de plantilla para sistematizar el proceso de registro de errores. El número de campos o celdas que contiene varía según el diseño y las decisiones de cada responsable de planificar el testeo. Estos campos deben contener información detallada para que otras personas identifiquen los errores y

⁵ «Los testers lingüísticos pueden identificar errores funcionales importantes [...] Es necesario que estos errores se registren y se comuniquen al desarrollador al máximo detalle para que puedan ser corregidos».

⁶ «Esto resulta de ayuda especialmente si los testers lingüísticos registran cualquier tipo de errores funcionales porque permite al desarrollador identificar rápidamente en qué versión del videojuego se encuentran dichos errores».

puedan corregirlos. Las empresas de localización y testeo crean sus propias plantillas y las envían a los responsables de hacer el testeo para que den toda la información necesaria sobre cada error.

If providing a standardized template for the translators to use, it is easier to keep track of the linguistic bugs across all the languages. The bugs can be merged into a single master spreadsheet and sorted in whatever ways are necessary (Chandler y Deming, 2012, p. 241).⁷

Como objetivo de este trabajo nos proponemos crear un modelo propio de *bug report* que contenga los campos necesarios para dar información concreta y detallada de cada error. A continuación presentamos la estructura de nuestro modelo.

6.2.2. Estructura

La estructura de nuestro *bug report* es una unión entre dichos parámetros, puesto que, algunos campos se repiten y como hemos argumentado anteriormente, se pueden registrar varios tipos de errores en un proceso de testeo, y registrándolos adecuadamente se logra su correcta comunicación a los responsables de corregir dichos errores. La estructura que proponemos parte de una tabla dividida en 10 columnas que contienen los siguientes campos o apartados:

- **Número de error.** Cada error posee un único número. Sirve para llevar cuenta del número de errores registrados o hacer referencias entre tipos de error.
- **Código de error.** Versión abreviada y codificada del tipo de error, su localización e idioma del que se trata. Con esta codificación se pueden aislar los errores de un solo tipo o de una sola ubicación o de un único idioma.
- **Tipo de error.** Con la finalidad de poder aislar los errores de un único tipo. Aunque conociendo el código del error ya es posible saber de qué tipo de error se trata.
- **Ubicación.** Descripción detallada de los pasos que hay que seguir para llegar al lugar donde se ha encontrado el error.
- **Descripción.** Descripción detallada del problema.
- **Texto localizado original.** Se añade el texto que contiene el error en el original, si procede, normalmente en los errores de tipo lingüístico. Puede ser de gran ayuda si el traductor subraya o destaca exactamente la parte del texto que hay que corregir.
- **Texto corregido.** Se añade la propuesta de corrección del traductor.
- **Estado.** Información sobre el estado del error.
- **Observaciones.** Si es necesario añadir cualquier comentario hacia el responsable de corregir los errores, hacer cualquier pregunta o responderla.

⁷ «Si se proporciona una plantilla estándar para los traductores es más fácil llevar un registro de los errores de cada idioma. Los errores pueden registrarse en una plantilla general y seleccionar diferentes opciones cuando sea necesario».

- **Imágenes.** Si es posible adjuntar capturas de pantalla. Se nombrará cada imagen con el número de error del que se trata, p. ej. «08.jpg». Si hubiera más de una imagen adjuntada a un error, la nomenclatura sería «08a.jpg», «08b.jpg», etc.

6.2.3. Modelo propuesto: ejemplo de plantilla de informe de errores

Núm. Error	Código de error	Tipo de error	Ubicación	Descripción
05	ES_INT_Mensaje bienvenida	Lingüístico	Inicio del programa tras la finalización de la instalación en Windows 8 de la versión ES	Al principio del mensaje aparece el término «computadora» y al final del mensaje la palabra «ordenador»
06	ES_Formato fecha_SL	Estético	Página principal esquina derecha inferior	La fecha que se muestra en la página de inicio no se corresponde con el formato adaptado a la versión localizada

Figura 1. Modelo de informe de errores con ejemplos.

Texto localizado original	Texto corregido	Estado	Observaciones	Imágenes
Este programa instalará Orbitron 3.71 en su <u>computadora</u> (...)	Este programa instalará Orbitron 3.71 en su <u>ordenador</u> (...)	Pendiente de verificación	El público meta de esta versión localizada utiliza el español peninsular como lengua vehicular, de ahí que se prefiera el uso de la palabra «ordenador» frente a «computadora»	05.jpg
Se recomienda cerrar todos los programas en uso en su <u>ordenador</u>	Se recomienda cerrar todos los programas en uso en su <u>ordenador</u>			
05/24/2018 (MM/DD/YYYY)	24/05/2018 (DD/MM/YYYY)	Corregido	Es posible que este error no se haya revisado en las versiones testadas anteriores	06a.jpg 06b.jpg

Figura 2. Continuación del modelo de informe de errores con ejemplos.

La propuesta de este modelo viene motivada por la necesidad de estandarización presente en los procesos de registro de información en ámbitos profesionales. En localización, las empresas utilizan sus propios modelos, que en ocasiones les vienen dados por los ingenieros de localización o desarrolladores y no se cuestionan su aplicabilidad y efectividad frente a una diversidad de proyectos.

Este modelo es estandarizado y aplicable a todos los tipos de testeo, a todos los productos localizados y a todos los idiomas, hecho que no elimina la posibilidad de modificación por parte de los usuarios en función de las necesidades concretas de cada proyecto. Se propone, por ejemplo, la posibilidad de ampliación de los campos o columnas para que permita añadir información sobre el revisor o *tester* al que está dirigido cada error o subrayados en diferentes colores para determinar la urgencia o importancia de cada error.

También es posible obtener metadatos sobre la frecuencia de aparición de cada error con fines de estadísticas internas e investigación sobre el error en localización, con el fin de aportar soluciones predeterminadas para los errores más comunes y poder prever su aparición.

7. El *test plan* o instrucciones de testeo

Determinar las pautas de actuación en cada proceso de testeo es esencial. Es conveniente que el responsable del testeo redacte un guion a modo de instrucciones que describa qué pasos se deben dar, qué áreas se deben testear, a qué pantallas hay que llegar y cómo se deben mostrar los elementos en cada una.

El *test plan* se presenta a modo de instrucciones que guiarán el proceso de testeo. El localizador seguirá estas instrucciones para realizar correctamente su tarea. Servirán de guía a lo largo de todo el proceso y si identifica cualquier tipo de irregularidad o incorrección, deberá registrarlo en el *bug report*.

El *test plan* debe contener también una serie de parámetros para el correcto registro de errores, qué se debe considerar error y qué no, explicaciones sobre la documentación anexa e instrucciones de formato. En este último apartado hemos propuesto un modelo de plantilla de un *test plan*, que, en cada caso particular, se podrá complementar con la información necesaria y que se adjuntará a una plantilla de *bug report*.

7.1. Estructura del modelo de *test plan*

Elaborar un correcto *test plan* es una tarea importante para el proceso de testeo. El encargado de realizarla debe detallar las pautas e instrucciones. Normalmente la realiza el gestor de proyectos de testeo, un perfil demandado en los proyectos de localización. La estructura que proponemos para elaborar un *test plan* es la siguiente:

- Información relativa al encargo
 - Fecha de inicio del testeo – fecha final del testeo
 - Responsable *testing* – Datos de contacto
 - Traductor/localizador encargado
 - Referencia del encargo
 - Idioma/s
 - Otra información necesaria
- Equipamiento necesario para realizar el testeo
 - Hardware/software*
 - Versión
 - Instrucciones de descarga
- Tipo de testeo que se va a realizar

Funcional/lingüístico/estético/regresión

- Pautas para la realización del testeo

Archivos

Formato

Localización en carpetas

Elementos que deben testearse

Rutas de acceso/pasos a seguir para llegar a cada elemento

Elementos que no se deben considerar errores

Documentación anexa

Observaciones

- Registro de errores

Bug report (Explicación de la información que debe registrarse de cada columna/campo de registro)

- Envío de archivos

Plataforma de envío

Formato: cambio de formato/formato original

Bug report completado en formato original

8. Conclusiones

En cuanto a los objetivos generales, hemos podido realizar un análisis del proceso de testeo y hemos presentado conceptos teóricos a modo de discusión sobre diferentes teorías obtenidas de una recopilación bibliográfica, que intrínsecamente ha conllevado al estudio de los parámetros utilizados en la práctica de este proceso, puesto que gran parte de la bibliografía es de componente práctico y hemos procedido a la abstracción de conceptos y teorías a partir de la práctica del testeo de *software* o el de videojuegos.

Por otro lado, los objetivos específicos planteados se han abordado con la taxonomía de errores propuesta y con el modelo de codificación para facilitar su registro e identificación en un informe de errores. Consideramos que esta taxonomía de errores es bastante concreta, puesto que recoge errores que se pueden dar tanto en localización de *software*, de páginas web y videojuegos.

También hemos propuesto, consiguiendo el resto de los objetivos específicos planteados, un modelo aplicable a la práctica de un *bug report* para la fase de testeo en localización; y que además completamos con ejemplos para comprobar su uso en contexto junto a una plantilla para completar las instrucciones o *test plan* a modo de encargo para el revisor, complementado así todos los objetivos propuestos inicialmente.

Consideramos que este trabajo supone una recopilación necesaria cuando hablamos del testeo en localización y logra enmarcarlo en los Estudios de Traducción, asumiendo la condición de investigación dentro del ámbito académico que pretende servir de ayuda a la futura práctica profesional y dejar las puertas abiertas a futuros trabajos de investigación que aporten nuevos matices y perspectivas, siendo latente el continuo avance de las nuevas tecnologías y, con ello, las posibilidades en el mundo de la localización.

Sobre la autora

Beatriz Bailén Ruiz es graduada en Traducción e Interpretación por la Universidad de Alicante. Ha realizado diversos trabajos de investigación, por un lado, sobre la gestión de proyectos y perfiles profesionales, nuevas tecnologías, didáctica de la traducción y formación de traductores en el marco la Beca de Colaboración del MECD en conjunto con el Instituto Superior de Estudios Lingüísticos y Traducción (ISTRAD) durante el curso del Máster Universitario en Traducción y Nuevas Tecnologías: Traducción de Software y Productos Multimedia que oferta esta institución. Por otro lado, con su trabajo de fin de máster se ha especializado en el proceso testeo en localización, proponiendo aplicaciones prácticas de los resultados y dando lugar a un modelo registrado como propiedad intelectual de la Unión Europea.

Agradecimientos

Gracias a Juan José Arevalillo por ofrecerme su ayuda, darme las pautas necesarias para realizar este trabajo y apoyar mi idea. Gracias a Anne Bécart por resolver todas mis consultas con detalle y estar ante cualquier problema. Gracias a todo el equipo del ISTRAD por estar cerca en la distancia. Gracias a los organizadores del III Congreso CIJITI por apostar por mi trabajo y permitir su difusión a la comunidad profesional e investigadora.

Referencias

- Arevalillo, J. J. (2012). La revisión profesional, herramienta de mejora de la capacidad de traducción. *Actas II Congreso Sociedad Española de Lenguas Modernas* (pp. 9-25).
- Chandler, H., y Deming, S. (2012). *The game localization handbook* (2nd Ed.). Sudbury (Mass.): Jones & Bartlett Learning.
- Esselink, B., Vries, A., O'brien, S., Vries, A., y O'Brien, S. (2000). *A practical guide to localization* ([Rev. ed.], The language international world directory, v. 4). Amsterdam: John Benjamins.
- LISA, Localisation Industry Standards Association (2003), LISA QA Model 3.0. <http://producthelp.sdl.com/SDL_TMS_2011/en/Creating_and_Maintaining_Organizations/Managing_QA_Models/LISA_QA_Model.htm> [16/11/2018].
- Luo, L. (2001). *Software testing techniques*. Institute for software research international Carnegie Mellon University Pittsburgh, PA, 15232(1-19), 19. <<https://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf>> [15/11/2018].

Miller, E. F. (1981). Introduction to Software Testing Technology. Software Testing & Validation Techniques. Michigan: IEEE Computer Society Press, 4-16.

Shanti, A. (2006). Localization and Internationalization Testing. 6th Annual International Software Testing Conference in India 2006. <http://www.qaielearning.com/KnowledgePapers/Localization_Internationalization_Testing.pdf> [17/11/2018].