@yolocalizo

CANARY SYMPTOMS IN SOFTWARE LOCALIZATION

- Localization is not a simple task as many people would think.
- Localization cannot be oversimplified as change words from one language into another.
- Proper planning can save us from many troubles due to poor preparation such as "the canary symptoms" indicated here.
- Perhaps the most expensive "canary mistake" we can make with software localization is letting "translations" sitting at the bottom of the Devs backlog to-do list.
- It's easy to think of "translations" as the last stage of production — but that's a costly assumption, assumptions that definitely would kill a canary..

EMBEDDING TEXT INTO SOURCE FILES

DIFFERENT LANGUAGES TAKE UP DIFFERENT AMOUNTS OF SPACE

NEGLECT THE IMPORTANCE OF THE STYLE GUIDE AND GLOSSARY LIST

CONCATENATED STRINGS

NOT PROVIDING CONTEXT FOR SOFTWARE LOCALIZATION

SEPARATE TEXT FROM GRAPHICS

EMBEDDING TEXT INTO SOURCE FILES

One common mistake is text being hard-coded into the files.
Embedding text in the code will affect our translations. It might be tempting to embed text directly into the code, especially if we'll release one language first. This is a bad idea and we must write code keeping in mind that the software might reach an international market in the future.

SOLUTION: Store all text as variables in a separate resource file.

NEGLECT THE IMPORTANCE OF THE STYLE GUIDE AND GLOSSARY LIST

Style guide and glossary list are essential for a localization project. They provide clear instruction for translators and it aligns expectations. With a good style guide, translators can adapt the content into local culture well, consistent with our brand style and tone.

SOLUTION: Invest time to create (and maintain) glossaries and style guides including different stakeholders expectations. Check out this post to know more about style guide best practices

NOT PROVIDING CONTEXT FOR SOFTWARE LOCALIZATION

We MUST provide localization kits and comments. The translator will not know as much as we know about our software functionality, the story of the video game we asked them to localize or the UI of the web page we outsource to them. The context for a translator is like breathing...

SOLUTION: Providing context to every string will help translators to fit top-notch translations

@yolocalizo

DIFFERENT LANGUAGES TAKE UP DIFFERENT AMOUNTS OF SPACE

When it comes to localization we must be aware that all languages do not take up the same amount of space. English strings usually are shorter than its respective translated version. A clutter UI with truncated keys might send the message that our app/company is amateur and the product has been done carelessly.

SOLUTION: Design for +50% and give keys room to grow and shrink dynamically

CONCATENATED STRINGS

Developers every now and then love to create concatenated pieces of sentences using placeholders. That's the perfect recipe to fail when we bring that content to a different language. That approach is based on a huge assumption: grammar rules and a certain sentence structure work equally in English than in other languages.; which is obviously wrong, remember the old saying of "When you ASSUME means that you make an ass out of you and me" :)

SOLUTION: Don’t assume grammar structure, don't concatenate placeholders and get into the habit that it’s best to create keys that are complete sentences.

SEPARATE TEXT FROM GRAPHICS

Complex bitmapped graphics are common to many critical translation bugs. Ideally, graphics should not contain text for the simple reason that it eliminates the need to translate it.

SOLUTION: If text must be associated with a graphic, we must create the text content as a separate component in our assets. Check out this tutorial to know more about how we can separate text from graphics using Adobe