Translation business management systems: online or desktop?

When choosing a business management system for their translation agency, a manager has to decide if that should be online or desktop. Each type has its pros and cons. Let's take a look at them.

Operating system

Desktop applications are installed onto a specific operating system. A program trading under a single name that is run on *Windows*, *Linux* or *MacOS* is essentially three completely different programs. And if, for example, all your project managers work on *Windows*, but you've decided you want to work on your beloved *MacBook*, you'll be able to get onto the system either if there is a *Windows* emulator installed on your *MacOS*, or if the system developer has provided for a variation of the program for *MacOS*.

Web applications remove this limitation. In fact, they don't need to be installed at all — they're already installed on a server. To access the online system all you need is a browser and an internet connection. You can manage projects from any device, and that doesn't just mean your computer, but your tablet and smartphone too. You don't need to find a computer if you want to check something on the database. Even when you're out at some translation conference, you'll always know how things are going with your project managers.

Technical support

There is no ideal program: errors occur in both types that sometimes catch the user out. The downside of web applications is that if the error is critical it stops working immediately for all system users. The upside is that fixing that error on a server means fixing it immediately for everyone.

It's more complicated with desktop applications; how they work depends on the specs of that computer (OS version, availability of alternate libraries, system settings, etc.). To correct an error, the support service has to ascertain what is causing the fault specifically on your computer. This takes time and effort for both you and the programmers.

Upgrades

Programs, including translation management systems, receive regular updates and improvements. Every upgrade eliminates any bugs that might have been found and adds new functionality. Generally speaking, upgrades need to be installed as soon as they become available.

With online applications, upgrades are installed without you getting involved, and even without you knowing about it: the next time you log on, you'll just see a new interface. With upgrades to desktop programs, you and your managers all have to download and install the application yourselves. In the process, errors sometimes occur due to the specific features of the operating system.

Integration with other systems

Online programs "talk" to each other across the application programming interface (API), coming together within entire ecosystems. For example, from a project created on a translation project management system, with the click of a button you can create a corresponding project on an online CAT system, and that is linked to a machine translation engine. Data automatically migrates from one system to the other, avoiding the need to manually enter the same information into different databases.

Desktop programs can also be linked to online databases, but they very rarely interact with one another: it is more complicated from a technical point of view. Their "talking" is normally a bit like manually performing the operation of importing and exporting files.

The way things are going, this class of isolated programs will die out in future: no one wants to waste time copying and pasting by hand.

It is even rarer for online programs to communicate with desktop programs. Online translation project management systems are hardly going to be able to access project databases on CAT tools installed on a computer. This means desktop applications have far fewer prospects for integration, and therefore for the future.

Operating speed

If the management system is installed on your computer, its operating speed will depend on the power of the processor, the amount of RAM and the operating system settings. In web applications, these factors are also important, but they are taken into account right from the outset, in the server parameters.

The speed of the internet connection can sometimes be a problem. Other than that, delays are possible at the server end if several users send an unexpected number of requests simultaneously.

The necessity of having an internet connection might also be considered a disadvantage: in contrast with using a desktop program, if you don't have an internet connection then working with an online system is impossible. But there are fewer and fewer places in the world where this is a real problem. That's why programmers for online systems don't even take it into consideration and consciously leave that niche for programmers of desktop programs.

Information security

This is the most controversial subject. People and companies are frightened of using web applications and storing data in the cloud, worried for the safety and confidentiality of their information. People assume desktop systems are more secure in this respect: if the data is physically stored on a computer, that means it's in your control. But is that really the case? Let's consider a couple of simple questions.

Is your computer connected to the internet? Unless you're working on a top-secret project in an underground bunker, the answer is going to be "yes". So that means that theoretically your computer can be accessed externally. And if you're not an expert in computer security, it will be easier for an attacker to hack into your computer than into a cloud service whose security is looked after by programmers.

Presumably you have to install updates for your desktop applications. Can you be sure of exactly what is happening on your computer when they are being installed? Theoretically, programmers could download your entire database while this is happening. The limitations to this are not so much technical as ethical and reputational.

Do you have a back-up copy of your data on another drive? If not, you risk losing it, because at any moment your hard drive could just crash. In the best case scenario, you'll be able to get a specialist to recover your data; in the worst, you'll have lost it forever.

With online services — especially translation project management systems — you get regular data back-ups. Copies are usually kept in a separate data center, to avoid the risk data loss as a result of the physical destruction of the data center by fire, flood etc.

On top of this, many people who are worried about storing data on the cloud have already been doing it for a long time. Most likely, your mail server will have a web interface, and that means your emails and files containing confidential information are *already* being stored on a server somewhere. It is also likely that you transfer files through online storage platforms like *Google Drive*, *OneDrive* and *Dropbox*. That is to say, in all likelihood you *already* store some sort of data on the cloud. The only question is whether or not you're happy to add that data to yet another system.

What it comes down to is that confidentiality is less a question of faith in the technology, than of faith in the programmers.

To simplify, the question of confidentiality can be split into three separate questions:

- Will the service programmers guarantee that my data won't get lost?
- Will they protect my data from hackers?
- Will they not steal the data themselves and sell it on to my competitors, or copy my business?

We have answered the first two questions above: the programmers will store and protect your data better than you can.

In answer to the third question: Imagine the scale of investment in developing a translation project management system. People spend years and hundreds of thousands of dollars developing this system, earning a reputation, investing in advertising, visiting conferences. And all this to kill the whole project in an instant by stealing the commercial data of a single company? This “operation” wouldn’t even cover the cost of creating the system, meaning it would make no sense economically. And it would be reputational suicide.

A serious business is a peaceful business. Programmers make earn money by gaining reputation and trust, and using this to attract lots of users, then getting paid by them for maintaining and supporting their system. In other words, it is in the programmers’ interests to protect the users’ data. It’s hard to say, even, who has a greater interest in protecting it — the programmers or the users themselves.

Conclusion

Desktop applications have the upper hand when you’re working within strict limits and don’t plan on stepping beyond them. Web systems are more flexible and easier to develop.

Desktop programs give you a sense of control and protection over data. But that’s just the way it seems: this is a legacy of the 1990s, when it was rare for any computer to be connected to the internet.

Link to original article