Educating Talents
since 1460.

University of Basel
Petersplatz 1
P.O. Box 2148
4001 Basel
Switzerland

www.unibas.ch

University of Basel

Getting started with **software literacy**

Educational Technologies | Project Digital Literacies
More information: https://digitalskills.unibas.ch
Contact: digital-literacies@unibas.ch

Version of this document: 2.1, March 2023

This document has been authored
by the Team Educational Technologies
of the University of Basel.

Getting started with software literacy

The world of software has undergone important transformations in recent years. On desktop/laptop computers, more and more programs are available as online tools, as opposed to programs to install on one's computer. On mobile devices, small programs called “Apps” have appeared that are literally at our fingertips and can assist us in almost any task. Due to these developments, programs nowadays seem easier to find, try and use than ever before. Moreover, many of them come at no apparent cost.

And yet, it is one thing to choose a tool for small, infrequent tasks; it is quite another to select software for serious work over many years – for instance in the context of one's studies or professional activity. Here more criteria must be taken into account, such as privacy, durability and interoperability.

This document presents the most important criteria to consider and give a few recommendations on where to search for software.

1

Criteria for software selection

Selecting software is highly dependent on how much we plan to use it. If we need an application to quickly solve an unusual problem (convert from a rare file format, for instance), then the fitness of the tool for the purpose, its cost and its ease of use are the main criteria to consider.

However, if we plan to use the program repeatedly over an extended period of time, it is advisable to take further criteria into account. The same applies when the data that shall be fed to the program is sensitive – even more so when the data is not about us, but about other persons.

When looking for a particular piece of software, we check first:

1. whether it can perform the task we plan to use it for
2. and whether it is compatible with our operating system.

The one being the reason why we look for software in the first place, and the other a necessary condition for us to use it, the two first criteria above are a sine qua non in software selection. However, several less obvious aspects are also important to consider. We now present seven of those.

③ **Security and privacy.** Some programs score very low in this area – in particular, of course, so-called “malware”, i. e. applications that contain malicious code (such as viruses or Trojan horses, which can take control of our machine). Less offensive, but also problematic, is “adware”, namely free tools that derive their revenue from displaying advertisement. In order to do so, they may gather user data in the background. Adware is far from uncommon in the world of mobile Apps.

It is advisable to take privacy and data protection seriously in all cases. Even in the absence of any legal obligation, being careful with data protection is a precautionary principle that can help prevent many risks that accompany the digitalization process (identity theft, unwanted targeting by ads, manipulation, attack by ransomware, etc. – for an introduction to this topic see, among many available resources, this New York Times article).

When working with data that concerns other persons, one is by law under an obligation to meet the legal requirements in this area. Full information in this regard is available in the online learning resource of the University of Basel entitled Data protection.

Things to check in the area of data protection are:

- ✓ The software can be installed from a reputable application store (stores check for malware – typically NOT for adware, however, nor for the gathering of user data).
- ✓ The program has a website, and it clearly describes the policy of the developers with regard to security and privacy.
- ✓ For mobile Apps, the program does not require permissions that are irrelevant to its function (e. g., a calculator doesn’t need access to our contacts).
- ✓ An internet search does not return warnings against the software at the level of security and privacy.

④ **Standard compliance.** A piece of software is said to be standard compliant when it implements internationally recognized norms and guidelines. This is especially relevant in regard to file formats: standard-compliant programs typically use generic file formats that can be read by many different applications, not just the one from which the file originates.

Using standard file formats increases the chance that the file will remain usable even after switching applications or changing operating system (including in the distant future when the original application has disappeared).

Things to check in this area are:

✓ The program's website clearly indicates a commitment to standards.

✓ The program uses standard file formats; if it doesn't, it is worth checking how easy it is to find applications that can reliably convert from the format.

⑤ **Development and support.** If we plan to use a piece of software over an extended period of time, it is obviously not advisable to choose one that is no longer actively developed.

In the short run, bug fixes, security updates and improved features may no longer be offered. In the long run, the program may stop running on our system, for instance after major system updates. Moreover, programs that are being actively developed usually come with up-to-date documentation, mailing lists and forums where users can find help and support.

Things to check in this area are:

- ✓ The program has been around for a long time (a few years of continuous active developments is a minimum).
- ✓ The last version was released recently (less than a year).
- ✓ New versions of the program are published at regular intervals.
- ✓ The developers communicate frequently about the program (for instance on the program’s web page or in a blog).
- ✓ The developers and/or the community of users provide good documentation and support.

⑥ **Resource consumption.** This is not necessarily the first thing we may think about, but it is also relevant. Some programs use more memory and processing power than others, and are correspondingly more resource-hungry. They may run more slowly, slow down other programs, and (on laptops) use more battery.

Things to check in this area are:

✓ User forums or specialized websites indicate that the software uses a reasonable amount of system resources.

⑦ **System independence.** So-called "cross-platform applications" run on different operating systems, including in some cases mobile devices. It may be advisable to choose cross-platform applications if we use (or may use in the future) different operating systems (for instance one at home and a different one in the office).

In this way, we make sure that we can easily work on our files anywhere. Note that all web-based tools are by nature cross-platform.

Things to consider in this area are:

✓ Apart from our own system, the application runs on at least one further widespread platform (such as Microsoft Windows, MacOS, iOS or Android.)

✓ (Alternatively) the application runs only on one system, but it comes with a web-based version.

8 **Cost.** Many popular programs, generic as well as specialized ones, are not free of charge. In certain areas, for instance multimedia creation, programs can even turn out to be very expensive.

We may be tempted to install so-called freeware instead, but these often entail reduced functionalities, contain advertisements, or gather data in the background. One option to reduce cost while also avoiding such problems is to look specifically for free and open-source software (FOSS).

This point is detailed below.

9 **Usability/learnability.** Usability is always a rather subjective matter. Basically, user-friendly programs allow users to start working productively with the program right away. This, in turn, depends on how "familiar" the program feels – in other words, it depends on how similar it is to other applications we already know.

None of these are especially good criteria to select applications, which is the reason why we suggest to focus not on the question of "familiarity" or "similarity with other Apps", but rather on the aspect of learnability.

Things to check to evaluate how learnable an application is include:

- ✓ The application is well-documented: guidelines, tutorials and help pages are easily available online.
- ✓ The application has a large, active user base that answers questions quickly, for instance in online forums.

2

Proxies for software selection

Admittedly, the criteria above represent quite many things to consider. In order to simplify the task of software selection, we may use one of the following proxies, or a combination of them.

① **Institutional support**. Organizations provide tools to their members which they have carefully selected according to the criteria above and for which they also offer support. This applies of course also to the University of Basel. Depending on your situation, you may benefit from a computer with pre-installed software.

On Microsoft Windows, you may also have access to the "Portal Manager", a tool that allows university members to directly install a selection of programs. (Press the Windows key and search for "portal".) The equivalent for MacOS is called "Managed Software Center".

Many tools are also licensed by the University of Basel for use by its students, for instance Zoom and the online version of Microsoft Office.

A presentation of the programs available to university members, as well as some good alternatives to them, can be found in the brochure "Getting started with alternative tools".

② **User base.** The size of the program's user base (in other words: how widespread, how "popular" it is) is arguably the primary criterion used by most people. The reasoning behind taking the program's user base as the main criterion is the following:

- A piece of software with a large user base is less likely to be abandoned by its developers (criterion 5).
- It is likely that information and help will be easily available, either provided by the developers themselves or by the community of users (criterion 5).
- The software is probably compatible with several operating systems (criterion 7).
- Its developers have an interest in making the software intuitive and relatively easy to use (criterion 9), as well as to optimize its resource consumption to make sure it runs on different system configurations (criterion 7).

On the other hand, popular applications may come with several inconveniences:

- They usually must be purchased (or, increasingly, rented) and their price is often high, especially when they dominate their market (criterion 8).
- They are not necessarily the best ones regarding security and privacy. Typically, none of them are malware or adware per se. Yet it is not uncommon for the company that stands behind the software to gather user data (criterion 3).
- They are not necessarily standard-compliant (criterion 4) – the developers may even have an interest in avoiding standards, especially regarding file formats, since this prevents users from easily migrating to alternative applications (the technical term for this is "vendor lock-in").

③ **Free/Open Source.** A third proxy is the FOSS nature of the software. FOSS stands for free and open-source software. FOSS applications are very widespread, but their success is especially strong on servers. In the world of end-user software, by contrast, they are less widely used. Yet in this area too they present a few advantages:

- To begin with, they can be downloaded and used free of charge (criterion 8), do not display advertisements, and their developers, as a rule, do not gather user data (criterion 3).
- They are typically more standard-compliant than their proprietary counterpart (criterion 4).
- Due to the availability of the source code, developers and enthusiasts can more easily adapt them to run on several different computing platforms (criterion 7).
- The development model of FOSS applications relies on a continuing interaction between the programmers and their users, who can submit bug reports and feature requests. As a result, patches and fixes are often more quickly provided (criterion 5).
- Likewise, at least in the case of bigger FOSS projects, there is a large community of users offering help and support (criterion 5).

Free/open source software also has downsides:

- While large FOSS projects, as suggested, are developed rapidly with frequent updates and bug fixes, many smaller ones suffer from a scarcity of developers, resulting in less features and rare updates (criterion 5).
- Because many FOSS applications have a smaller user base, users are less likely to find help in their own circle of colleagues and friends (criterion 9).
- FOSS applications have the reputation of being more complex, more technical, and less user-friendly overall. Due to various factors that cannot be detailed here, their “look and feel” is less familiar. Learning the tool may be, as a result, a bit more complicated.

3

Where to find software

Integrated software repositories

- Some systems, especially iOS and Android, have integrated software repositories, better known as “App Stores”. On Android, additional repositories can be added. For example, F-Droid is an alternative repository that offers only free and open-source software.
- Newer versions of Windows offer a wide range of software in the Microsoft Store.
- As mentioned, organizations can also maintain internal software repositories. On the computers of the University of Basel running Microsoft Windows, programs can be installed over a small utility called the “Portal Manager” (Windows) or the “Managed Software Center” (Mac).

Online software repositories

- For open-source software, developers’ websites such as Github or Gitlab offer a huge amount of resources: source code, documentation, forums, and programs as downloads.

Specialized websites

- The website G2 is a popular starting point when looking for business software.
- When looking for a replacement for, or an alternative to, a particular application, the crowdsourced website AlternativeTo can be very useful.
- The website opensource.com, supported by the free-software company Red Hat, contains many useful pages with descriptions and recommendations for open-source software. The page Open-Source Alternatives, which lists FOSS versions of many common applications, is a good starting point.
- The EduTools Portal at the University of Basel contains a lot of information on the programs and web platforms used for learning and teaching at our university. The university offers support for all applications listed there.
- For information on safety and privacy with mobile software, the web site Mobilsicher is useful; another site with good software recommendations for the privacy-conscious is PrivacyTools.