Computer Software

In the modern world of computers and electronic devices, there are many types of 'software', these include:

- Desktop – software designed to be stored & accessed on a desktop or laptop computer, creating data that will be saved either to that computer or an accessible computer or server on the same network.
- Browser – software accessed via a web browser on a desktop or laptop computer, or an electronic device such as a Tablet or Smart Phone. Data will be stored on a remote server linked to the provider of the browser software.
- Apps – software designed for Tablets and Smart Phones. Depending on the App and its purpose, data may be stored on the device and/or a remote server linked to the app provider.
- Hybrid – software that provides access (to functionality and data) on both desktop/laptop computers and smart devices. The version for the computer can have both/either desktop or browser access. Depending on the software, its data may be stored on a network server or a cloud server.

Purchasing software for business purposes, its origins may be:

- Off the Shelf – a pre-existing package of software. It is purchased from a vendor "as is". You have no control over its features beyond user-definition utilities that may be provided.
    - o Outright purchase. No further financial outlay is expected or mandatory. Other products such as support, add-ons or upgrades may be offered. While related, in that it isn't relevant without the initial purchase, these are separate purchases in their own right.
    - o Subscription purchase. This is quite common for Browser based software. Access to the functionality is dependant on continued payments.
- Customised Software – you discuss your needs with a developer and they provide software to fulfil those needs.
- Combination – a software developer starts with a 'base' package and adds your requirements to it.

Logical Developments provides Desktop & Hybrid software as an outright purchase, in the following formats:

- **Off the Shelf packages.**
- **Off the Shelf packages based on our modular system.**
- **Customised Software Solutions, either in full or using our modular system as a starting point. Your requirements can be added as new features, or a customised module can be developed.**

"When buying software, what am I actually purchasing and what rights do the vendor and purchaser have?"

When you buy software, whether it is off-the-shelf or a customised solution, you are in fact, buying 'the right to use it'. This is why software comes with a 'user-license'.

Software brings challenges because you are buying something that doesn't have physical quality about it. In the modern era, it is downloaded from the Internet so you don't even get a manual.

Obligations and rights:

- The purchaser has the obligation to ensure the software will address their needs. This is true for both off-the-shelf and customised software. For customised software, the onus is greater as the purchaser has had direct input into the features that are developed.
- The software provider has the obligation to ensure the software does what advertising, manuals and/or specifications, says that it does.
- If the software isn't doing what you want or expected:
    - o <u>Outright purchase</u> - you need to negotiate with the vendor for a refund according to Consumer Affairs guidelines. This is difficult, as you don't have a tangible item to 'return in saleable condition'. It is also challenging to show that it hasn't been used and/or is completely uninstalled. The available argument is that it is 'unfit for purpose'. For example, a calculator app that doesn't add up 2 + 2 to equal 4.
    - o <u>Subscription purchase</u> - you can simply stop paying the subscription and/or negotiate for a refund. Note – if you have been using it for a time, it is most likely that you will be cut off from your data.
    - o <u>Customised Software</u> – as you directly contribute to the features that will be provided, "unfit for purpose" is only arguable if it does not do what the specification said that it would and, that by mutual agreement, it is not fixable.

Last Updated: 16 March 2023

- The vendor has the obligation to refund an off-the-shelf purchase where it isn't fit for purpose. If it is customised software, they have the obligation to fix it. If the software works as advertised, the vendor does have the right to refuse a refund.

It is very important to read the Terms & Conditions, Warranty Policy, Contracts and Manuals very carefully. These will outlay exactly what you are purchasing. For customised software, your contract - that may be in the form of a quote and/or specification - will set out what you are purchasing and what each 'milestone' payment is covering. If you change your mind part way through the project, you most likely won't be due a refund for payments made so far. Eg it pays for the preparation of the Specification document and you have received it. Ie you have received what you paid for.

What you are not buying:
You are not buying the 'rights of distribution' to the software. Ie you cannot use the software, duplicate it and sell it to someone else. This is distinguishable from buying a physical thing, like for example, a box. If you buy a box, you can use it and when you no longer have use for it, you can give it away or sell it. It is impossible to both keep using it and sell it.

The exception to this is on-selling software as part of something bigger. For example, you are selling your business as a going-concern. The computers and software are on-sold as part of the business. In this case, you do not have the right to keep using said software on your home computer (unless of course, you had purchased a separate license for home use).

With customised software, for example, you have an idea for an app – if you want the sale rights of that software, this requires a special contract with your developer. It will make it more expensive. Most developers provide software at a reduced rate as they retain the resale rights and once developed, they can sell it again for a return on their investment of time. (These terms and conditions should be outlined in the Contract/Quote/Specification.)

Development Life Cycle

A typical software development process at Logical Developments happens like the following:

1. Meet to discuss requirements
2. Quote provided including:
   - Draft mock-up and summary of requirements
   - Payment schedule such as: 40% acceptance deposit; 45% delivery, 15% customer acceptance. (Larger projects may include more progress payments signified by recognisable milestones.)
3. On acceptance of quote, a Specification Document is prepared.
   - Email or phone LD to accept the quote
   - A 40% deposit invoice will be issued. Once the Specification is complete, this is non-refundable as it pays for the Specification document.
   - When paid, the preparation of the Specification will commence.
4. The Specification document outlines the requirements and how they are going to be addressed. Where possible, illustrations are provided to display how the software will operate. It is important for the client to read and understand this document. If anything is unclear, or you don't agree that it is correct – contact your developer. It is easier to change a Specification than to rewrite software.
5. If required – the Specification and quote (and payment schedule) are updated.
6. Accept the Specification by an authorised person signing the front page and returning it to the developer.
7. The software will be coded according the Specification.
8. When completed, it will be installed on site with a time-limited demonstration license. And the 'Balance on Completion' invoice will be issued. On-site testing and the warranty period will commence.
9. When the 'Balance on Completion' invoice is paid, your purchased software license is installed.
10. Warranty period: if a specified feature is missing or not working as described, it will be fixed free of charge. (Please see Logical Developments Warranty Policy.)
11. Additional requirements: (ie outside the scope of the original specification document) it is quite common for new needs to be identified once new software is installed. Or over time, needs change. Depending on the size of the new requirement, this can be addressed under a Support Agreement with phone/email correspondence, or commence a new development cycle.