

Creating a Software House

by John McMillan

Creating a Software House

Selling Software as a Product

Sometimes organisations develop computer software which they wish to sell. Other organisations will need to diversify into computer products. For example the author has advised a publisher to produce I.T. based products rather than paper ones. However, software sold as packages has different needs to bespoke software. Also, the selling organisation will be unlikely to have experience in the software market. This article describes issues that need to be addressed.

Design

A major difference between package software and in house bespoke software is that the latter needs to be more generic. If you decide to market software that was developed for your own use, you will need to address this. The software probably started with a tight specification that reflected your needs. However, not everybody will work the same way as you. There will always be variations, perhaps minor in your new customers' needs. Building flexibility into the programs will increase the market.

One of the first steps is to ask the ways in which other parties may work differently to you. There will be some difference in processes. They may use more or less skilled operators than you. Your input and control systems may have been designed to suit your staff. Different customers will want different levels of security. Customers may need to interface with other systems, both computer applications and manual. Managers may need different reports.

The computers themselves may be different. You may have designed the software to reflect your own file store. Some customers will want to install on single PCs, some on networks. They may have a mixture, with different network mapping on different PCs. The package software will need to cope with this. Screen resolutions vary. What looks fine on one screen may not work well at a different resolution. Printers differ, and printer drivers are notoriously prone to problems.

Trying to sell a program that rigidly fits one party is a recipe for disaster. It is vital to find out what may change between users and apply a means to apply these changes. Ideally the user should be able to configure the software himself. The easier this is to do, the lower will be support costs.

Computer Platforms

In house systems tend to be written for single computer platforms, for example Windows 98 with Laser Jet printers. However few software houses can get away with supporting a single platform. Your customers will demand software that runs on their current platform. Only major software vendors can afford to cut off customers with different platforms. Not

Creating a Software House

all computer users upgrade their entire IT at a stroke – most phase in upgrades. This means that even within a single client you may have to support a range of platforms.

Questions need to be asked. Should the software be restricted to, say, Windows? Is it even feasible to do otherwise? How quickly do our customers upgrade to new versions of Windows? How long do we support old versions?

Testing of multiple platforms has to be considered. How much testing has to be platform specific? How do you provide every platform?

Quality and Control

As the development team grows, version control becomes a problem. There will be a delay between writing the programs and distributing them to customers. During this time, the development team often start working on the next version. It is vital to keep copies of the “signed off” version as well as the working copies. This results in multiple copies and versions of the source files.

Often, developers take copies of a master source file and work on that. Without great care, two programmers could each make copies of one file and modify it separately. This is disastrous. Version Control software can be installed to guard against this.

Logging of bugs, other user problems and change requests has to be planned. What works for a small number of in house users will not work for a large customer base. A neglected in house users will complain loudly, a customer will take his business elsewhere. This is discussed in more detail under support.

Test policy must be designed. Fixing problems in house is relatively easy. Fixing problems when the software is in use by dozens or hundreds of customers is not. At the least, you will have to send out numerous upgrades and take many support calls. You will also appear in a bad light. For this reason, a comprehensive test schedule is needed. Most software houses then send the software to a small number of close customers. This is called the alpha test. After the alpha test has been completed, and ensuing problems fixed, the software is sent to a larger group. This is the beta test. Only after fixing the problems exposed by the beta test is the software sent to all the customers.

How to Market the Software

An IT product will often be a new venture for a company. It is too easy to underestimate the time taken to sell it. Existing products will have been sold for a long time. There will be many sales in the pipeline, there will be repeat business. Sales staff will have a good knowledge of the products and will be comfortable selling them. When an IT product is launched, all this changes. There will be few if any prospects and no repeat business. The customers may be different. Even present customers may have very different buying processes for IT systems.

How should you time the marketing? Clearly you do not want orders before the product is ready. But it is necessary to be realistic about the buying cycle. If this is 9 months (not

Creating a Software House

unusual), should you really wait until the product is fully ready before telling anybody about it?

The first contacts with prospective customers will be demonstrations. In the early days, a demo will be a two way process. You will show the prospect what the product does and he will tell you about areas that need changing. Perhaps you could treat early demos as market research.

You will certainly need some reference sites, customers who are prepared to tell their peers about your product.

Demonstrations

Demos need careful planning. They need to show the major features of the package but also demonstrate its benefit to the user. They must be scripted carefully. One approach is “a day in the life of our customer”, another is to show a case from start to completion.

There are broadly two approaches to demos. Your sales staff can carry them out or you can send demo versions to the customer. The first has clear advantages. You are in control, it is easier to get across your message and to deal with objections. However, on site demos are expensive and unlikely to be viable for a product costing less than £2000. The alternative is a CD which is sent out with a marketing pack, or a demo which is downloaded from the Internet.

Support

This is vital. It has to be accepted that users will find problems. Some of these will be caused by program bugs but there are numerous other reasons for support calls. These include

- Lack of understanding of the software
- Need of features that are not present
- Installation problems, difficulty interfacing to their PCs
- Receipt of faulty disks
- Advice on use which is beyond the scope of the documentation

Staff must be trained to react to support calls. Callers tend to be worried. This leads them to become aggressive. Support staff must know how to calm them down. They must be trained to talk users through problems.

Support issues must be prioritised. Some need urgent resolution. For others, work rounds can be found which allows a final correction to be scheduled.

All support calls need to be logged and recorded as live until resolved. At all times it must be possible to know what support issues are outstanding. When bugs are found, these must be placed in a bug log and scheduled for correction. Misunderstandings must be logged – these point to weak areas in the documentation.

Creating a Software House

A positive attitude to support is important. Much of your contact with customers will arise through support. Your handling of it will affect the customers' perception of your company more than any other factor. Support should form a major part of your customer relationship relationship.

Finally

The gains from selling software packages can be large. However, an organisation has to be put in place to manage this properly.

Other Articles

[Basics of Starting a Business](#)

[Basics of Finance](#)

About John McMillan

John McMillan is an interim manager specialising in high tech start-ups and marketing. He gives many workshops and classes on these subjects.

John has many years experience running software companies and launching software products. Some of his [projects](#) and more [articles](#) are described in www.mcmillantech.co.uk. John can be contacted on <mailto:john@mcmillantech.co.uk> or 01787 371099.

