In the first of a six-part series on software sourcing, **Ray Murphy** clarifies the main challenges to purchasing professionals in this complex area

SOFTWARE SOURCING

Beware the digital traps

The aftermath of the dotcom crash and, more recently, the global economic slowdown, have severely affected IT investment worldwide. Budgets were cut; projects cancelled or put on hold. In the software arena, many companies went under or were bought up as their share prices plummeted.

However, in the midst of this doom and gloom, companies worldwide continued to spend money on IT – albeit not so generously – as they realised that failing to invest in their business systems would put them at a serious disadvantage.

The effect of that continuing investment is now becoming evident. The US Department of Commerce reported that PC sales had jumped by more than 150 per cent in the third quarter of 2003 (ending last September) and that software purchases were a "major contributor" to the rapidly improving economy.

Software company stock market crashes are not the only hazard that software buyers have to contend with, though. Of all the products and services procured by large organisations, software must rank among the highest-risk categories.

Although the potential business opportunity resulting from the implementation of software applications is significant, software horror stories are common. The London Stock Exchange Taurus system reportedly overran original projections by £400 million, and the London Ambulance Service's system failure also cost millions of pounds.

And when the Passport Agency's new computer system failed in 1999, the political fallout of voters missing their holidays reached the highest levels of government.

The Standish Group's 1995 Chaos Report found that more than 31 per cent of projects were cancelled before completion and just over half overran their original estimates by more than 180 per cent.

The report also found that only 16 per cent of projects were completed on time and within budget. It is estimated that of the 300,000-plus software projects under way in the US at any time, between one-third and two-thirds will exceed the agreed delivery schedules and overrun.

The challenge for the software buyer is not just to bring in the right deal at the right price, but to be aware of and manage the significant risks associated with software purchasing.

With a major customised systems project, for example, how can a software buyer ensure that the supplier will be focused to deliver what is required, on time and within budget? How will the deployment and use of the software be managed once it is installed? What rights will the buyer have if the software fails to meet the defined specification or acceptance criteria? If a software company is sued for infringing a competitor's intellectual property rights, how will the buying organisation be protected against a similar claim?

This is only a subset of the risks and issues to be dealt with by the software buyer – the challenge is how.

On the street: the failure of the Passport Agency's new computer system in 1999 forced applicants to queue and some to miss their holidays

PHOTOGRAPH: PA

Massive overcharging

In a recent keynote address at the SunNetwork Conference in San Francisco, Scott McNealy, Sun Microsystems' chief executive, told his audience that software companies were overcharging by a factor of 10. Successful software companies post significant year-on-year increases in revenues and profits, even during recessionary times. Microsoft was founded in 1975 and within 25 years had established itself as the largest software company in the world and turned its founder, Bill Gates, into the world's richest man.

Businesses are aware of the potential that software has to differentiate them and create a competitive advantage. This can be manifested in better business products or services, or reduced operating costs.

However, software suppliers are aware of this and tend to capitalise by charging significant software licensing and support fees, and by imposing restrictive usage constraints. Once an organisation has bought the rights to use the software, it tends to stick with it because of the heavy switching costs resulting from the customer's investment in the software, the effort and cost of training its personnel to use it, and the cost of integrating it with other software systems, both inside and outside the organisation.

It is imperative that the people whose role it is to select and procure software are as well versed in the field as possible. This applies not only to the professional buyers but also to the internal technology and business personnel who are involved in the project. However, because of the intangible nature of software and associated services, it is often not clear exactly what is being purchased or what it is going to cost. Part of this perception may be due to the fact that software is quite intangible: in its basic form it cannot be seen or felt.

How the software is supplied is also different to other products or services. The customer licenses the right to use the intellectual capability that is embodied within the software, but does not usually own it and is generally constrained in the manner in which the software can be used. There is also no standard software licensing model because software companies tend to devise their own increasingly restrictive licensing regimes to maximise their income.

Every day, multi-million-pound software licensing, bespoke development, support and systems integration agreements are being implemented, often on the back of poorly negotiated commercial and contractual terms. Often this is not recognised until a project has gone wrong, when it is invariably too late.

While there is no panacea to cure all software project ills, the early involvement by skilled purchasing professionals, working closely with their business and technology colleagues, will almost certainly result in better commercial agreements that will, in turn, provide software suppliers with the incentive to deliver on time and within budget. This will then help to deliver the business benefits from the implementation of new or upgraded software systems.

Unlike other commodities, the licensee of a software product does not have any implicit rights to pass on the software to another entity, whether it is another department within the organisation or a business within the group.

Therefore it's not unusual, particularly with large systems software, for software companies to seek payment each time the software is moved, regardless of whether the customer is enjoying additional use or benefit from the software. Needless to say, this poses significant exposure for such business customers.

One of the few ways in which this exposure can be addressed effectively is by taking it on at its source – the selection and purchasing stage. The gauntlet has been thrown down to the purchasing community and, in many instances, it has responded well.

However, because software exists in so many guises and can, as a result, be a complex and intangible proposition, it requires the involvement of experienced, professional buyers to buy well. Non-professional buyers, such as IT staff, are often at a disadvantage when negotiating and implementing software agreements.

● NEXT ARTICLE (22 January): sub-contractual agreements

■ Ray Murphy is a specialist consultant in software sourcing and procurement (www.strategiccomputing.ie)

SOFTWARE DEFINITIONS

A multitude of packages

Within large organisations, the task of classifying and differentiating software types is often a challenge in itself. For end users it is generally termed "application software", such as a HR system, an accounts payable application or an Internet or intranet-based e-procurement system.

The software used to create the application software is called "development software", such as the programming language Cobol; compilers and assemblers take the "source code" in which the application is written and compile or assemble it into machine-executable "object code".

This executable object code interacts with the operating system software (for example, Windows XP on Intel PCs, Sun Solaris on Sun workstations, OS/400 on IBM AS/400 systems and MVS on IBM S/390 mainframes) and the "operating system" software interacts with the hardware or chip manufacturer's "firmware" – essentially a set of machine instructions that perform operations at the most basic level – in order to run the application as experienced by the software's end user.