Who Wrote This $%&?! Program?

part I

Quality software: does it exist and how do you know? What use is ISO 9000 in the software engineering process? Why should we audit software vendors? What happens if the software application is great but the company is not? In the first part of this two part article we discuss the benefits of performing a software audit and whether current regulations and guidelines are appropriate.

R.D. McDowall

It usually happens when you are in a hurry to produce those results or analyse the data you promised your customer last week. Working at your computer terminal you get one of those familiar and well-loved messages on the screen:

```
General Protection Fault at 1643EF
Enter any 11 digit prime number to
continue
```

Almost all of the software we use has been written by someone else. Do you ever wonder about the quality of these applications? Following on from last issue's discussion on the need and rationale for a user requirements specification (URS)[1], we will look at the other side of the coin at the vendors of software and the process of defining, developing and supporting an application.

We briefly discussed the software development cycle in the last issue when we looked in detail at the URS: the reasons for writing it, and the importance of doing so in understanding what you want the system to do. The requirements will be used as the basis of the tests to select which application you will purchase.

Why should you audit vendors? Trust but verify

The next stages of the software life cycle: the design, build and test phases, will normally be undertaken by the supplier or vendor of the software. Unless you are involved in the development, or the specification of the application, you will have virtually nothing to do with this part of the life cycle. Here comes the problem, how do you know that:

- the application that you want to buy is any good?
- the procedures for developing and supporting are quality ones?
- the application will continue to be developed?

Sometimes you may be given assurances by the salesperson and possibly a letter of intent, but an understanding of the company and its processes may not be available.

Therefore, the main emphasis of this article concerns vendor audits and will discuss:

- what is an audit?
- why should you audit vendors?
- should you audit all companies?
- what should you do during an audit?
- what happens if a company fails an audit?

The overall process is shown in Figure 1. The process, running in parallel with the definition and selection of the vendor, is to:

- define the scope and boundaries of the system or application

- assess the business and regulatory risk of the system
- make a decision: if business and regulatory risk is high then a vendor audit is required, if low, then no further action is required
- notify the selected vendor of your intention to audit, if they refuse then select another supplier
- agree a date and send the audit checklist
- conduct the audit and report results
- decide if the vendor is acceptable and purchase the system, if not either select another vendor or start a process of supplier management.

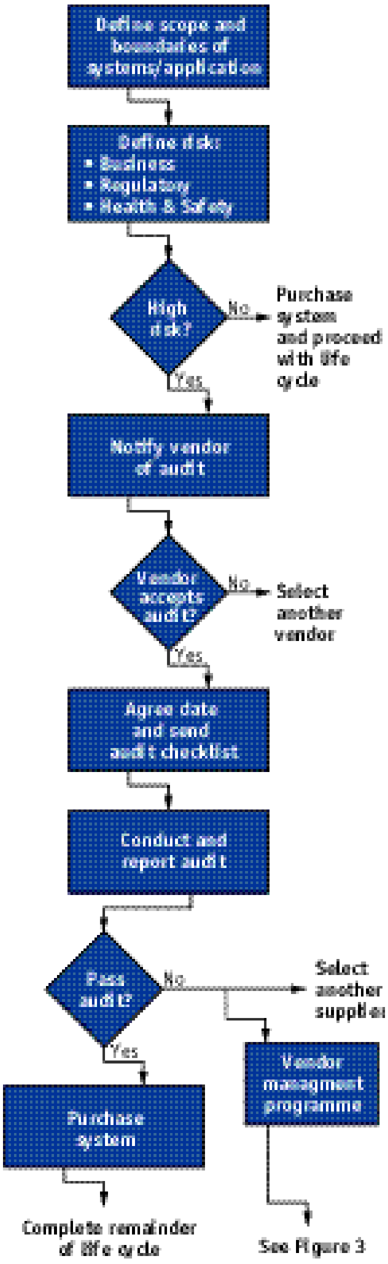

figure 1 Overview of the vendor audit process.

Your overall purpose in a vendor audit is to find out the quality of software development and what you, as a user, should do to ensure that the system selected and the company that supplies it are suitable. The overall philosophy that I would like to present is that the software vendor should become an extension to the laboratory and a business partner. However, it is a two-way street: what the vendor should do for the purchaser should be reciprocated by the purchaser to the vendor.

What is an audit?

An audit is essentially an independent check of a service or a product (here a product can be defined either as a unit of work or as a finished product). There are three types of audit: first-party, second-party and third-party audits.

If the audit is done from inside an organization it is called a first-party audit. Normally, this is achieved through either a separate quality-assurance group specifically established for the role, or by using part-time staff from other areas who are independent of the area being audited. The focus will be either on the quality of work or to see if written procedures have been followed, or possibly both.

A second-party audit is performed by a customer or somebody on their behalf, who will assess the vendor and their ability to design, produce and maintain a product or service. This is the class of audits that we will spend most of the time discussing in this article. Audits on behalf of customers can be applied to suppliers of raw materials, components or software; practically anything where the quality of the product can affect the operation or output of the customer. Obviously, in this article we will examine software only.

A third-party audit is through an independent accreditation body using guidelines published by the International Standards Organisation (ISO) or its equivalent. This results in the vendor being certified under a particular quality scheme. Furthermore, this type of audit is ongoing with surveillance visits every 6–12 months and reassessment visits every 3–4 years, depending on the quality scheme.

Obviously, the further removed from the organization that did the work the more objective the results. However, the intimate knowledge of the individuals and the processes involved is much less. Therefore, when considering a vendor audit we are looking at a second-party audit by yourselves or your representatives, such as a consultancy that specializes in this work. This may be backed by the use of third-party audits through accreditation bodies and their agents.

The principle I would like to discuss is how far can you take work on trust, especially if you work in a regulated industry, and how much should you verify through an audit. The short answer is that you have to use judgement. The aim of any purchase is that you make the supplier of a product or a service an extension of your organization. The aim is for a long-term, mutually beneficial, relationship. This is true if you want the latest version of Matlab or Excel or a scientific database management or control system.

When do I audit?

Before you purchase is the short answer. In the “honeymoon” period between selecting its system and placing an order the vendor can be very helpful. This may change after the application or system has been delivered. Therefore, if there are any corrective actions that require money holding back to ensure they are done, it is best to get them onto the table as early as possible in the process. It is difficult to recover from a poor bargaining position as it will invariably involve more time for discussions and/or legal action.

There is an exception to this rule, and that is when you are undertaking a retrospective validation of a system. If the system is crucial to the operation of the laboratory with a long lifetime, then a vendor audit is highly recommended. However, if the system will be replaced in the short-to-medium term, it would be better to put the resources into a more rigorous validation of the replacement system.

In parallel to the vendor audit, you should review the vendor's contract, and, as a result, you may also want to negotiate some terms in the contract. Combining the two is good timing and good sense.

Assessment of business, regulatory and compliance risk

There are many software packages available today, and the question is which software vendors should you audit and why? Look at the package you are purchasing and its function. What is the impact of the system on your organization?

- Is it an application that is only used for a single use with little impact, or is it a large system with a greater impact in terms of time or money lost through production delays if the system is unreliable?
- Are there health and safety implications: production control systems that must handle dangerous reactions or the system holds data that is valuable for your organization?
- Is the system covered by regulations or compliance issues? Does the system directly affect the quality, safety or efficacy of a product?
- Does the system store or manipulate data or information that is used to support patent applications?

It is the larger systems or those applications with major impact on your organization, whose vendors you should consider auditing.

For these critical software applications, the approach is taken on trust with verification of the process through a vendor audit. The phrase "trust but verify" comes from the disarmament process of nuclear weapons between the former USSR and the USA. Here the actual destruction of the nuclear weapons was made open to inspection to the other side either by on-site inspections or monitored from afar by the use of spy satellites.

What do regulations and guidelines say?

To help the risk-assessment process, a number of industries, such as agrochemical and pharmaceutical, and organizations involved with environmental analysis and monitoring, are subject to compulsory regulations, such as Good Laboratory Practice (GLP) and Good Manufacturing Practice (GMP). In addition, there are voluntary guidelines, typically ISO Guide 25, which are applicable to calibration and testing laboratories. Regardless of the industry or regulation or guideline, the main thrust is towards equipment that must be fit for purpose, properly installed and qualified. A typical example is:

European Union Good Manufacturing Practice Regulations Chapter 3.41

> Measuring, weighing, recording and control equipment should be calibrated and checked at defined intervals by appropriate methods. Adequate records of such test should be maintained[2].

There are similar statements from every other regulatory agency worldwide and ISO Guide 25.

However, when it comes to computerized systems and software used in the laboratory, there is more detail in supplements to the GLP and GMP regulations and pharmaceutical industry guidelines:

- Organization of Economic Co-Operation and Development (OECD) GLP principles for computerized systems state, "For vendor-supplied systems it is likely that much of the documentation created during the development is retained at the vendor's site. In this case, evidence of formal assessment and/or vendor audits should be available at the test facility"[3].
- European GMP Annex 11, section 5 states that, "Software is a critical component of a computerized system. The user of such software should take all reasonable steps to ensure that it has been produced in accordance with a system of Quality Assurance"[2].
- Good Automated Manufacturing Practice (GAMP) guidelines, May 1996[4] has Appendix C dedicated to supplier audits.
- Good Automated Laboratory Practice Guidelines from the US Environmental Protection Agency[5].

Therefore, the rationale for a software vendor audit in regulated industries comes directly from the regulations applicable to the industry.

However, with voluntary guidelines, such as ISO Guide 25[6] there is a different picture. There is still the need to demonstrate fitness for purpose; however, the detailed steps necessary are not stated or even implied. For example, NAMAS M10 accreditation standard (the UK interpretation of ISO Guide 25)[7] states in section 6.11:

"Where computers or automated test equipment are used for the collection, processing, manipulation, recording, reporting, storage or retrieval of calibration and test data the Laboratory shall ensure that, where applicable, the requirements of 6 of this Standard are met. The Laboratory shall, wherever possible, ensure that computer software is fully documented and validated before use."

Good at a general level but there is nothing in detail to help the laboratory implement this. However, is help at hand? The Guideline for Accreditation for Chemical Laboratories from WELAC[8] interprets ISO Guide 25 for computers in two sections: Section 10 on the use of computers, and Appendix C. However, the advice offered here is naïve and vague in the extreme. The reason for my statement is that there is:

- no concept of the role of the user in defining or documenting their requirements;
- no concept of the whole of the system development life cycle; and
- only considers "validation" as testing when the system is installed at the user site.

Whilst the use of validation ISO/NAMAS is partially consistent with the Institute of Electrical and Electronic Engineers (IEEE) definition of validation ("process of evaluating software at the end of the software development process to ensure compliance with software requirements"), under ISO/NAMAS there is no need for the laboratory to define its needs in a user requirements specification. This is compounded because there is no mention of the parallel requirement under IEEE of verification ("process of determining whether or not products of a given phase of the software development cycle fulfil the requirements established during the previous phase")[9]. The advice offered in the WELAC document is best described as similar to the point at which the pharmaceutical industry was in the mid-1980s and can thus be considered of little extra help compared with the original. The approach taken here only goes to show that quality in software cannot be tested in, it must be designed in from first principles.

Therefore, from an ISO Guide 25 perspective, a vendor audit and user requirements specification are not considered. My advice, based on experience, is that you get what you do, or do not pay for. Therefore, writing user requirements specifications and performing vendor audits are both common sense and essential for investment protection.

ISO 9000: Saint or sinner?

Many of the companies that you could audit will be ISO 9000 certified. Is it worth auditing these companies? They have a quality system in place and produce a quality product. Don't they?

There is much made of ISO 9000 accreditation and certification, especially by vendors, as it promotes a quality philosophy within an organization. The philosophy behind ISO 9000 is to document procedures and processes to ensure that they are adequately controlled and that output is consistent. This is similar in many ways to many other quality schemes, such as GLP, GMP and ISO Guide 25.

It is important to note that ISO 9000 does not guarantee product quality. The underlying principle that ISO 9000 is based upon is that organizations that follow documented practices and procedures in a consistent manner are more likely to create products that meet the customer's needs than those organizations that do not follow accepted practices and procedures[10].

My sceptical view of ISO 9000 is that it produces a poor product with bad processes that are well documented. You can never buy a software application, or any product for that matter, blind, based only upon ISO certification of the company, as we shall discover later in this article. Hence, the importance of the user requirements specification in defining the application you want, followed by time spent evaluating and selecting vendors in the market-place.

Let us explore in more detail the two key elements for ISO 9000. The first is the quality manual and the associated documented procedures it covers, the second is the scope of certification open to vendors.

The ISO Quality Management System is universal to all ISO schemes and covers four overall areas:

- quality policy statement;
- quality manual with overviews of areas such as organization, roles and responsibilities, training records, quality function, customer complaints, etc;
- written and authorized procedures detailing how the policy and manual will turned into an effective system; and
- internal audits by the quality manager or representatives.

Flavours of ISO 9000

There are three main types of ISO 9000 certification covering the whole, or just part, of a product or service life cycle:

- ISO 9001: quality assurance in design, development, production, installation and servicing. Conformance to specified requirements is to be assured by the supplier throughout the whole life cycle of a product or service[11].
- ISO 9002: quality assurance in production, installation and servicing. Conformance to specified requirements during production, installation and servicing[12]. (Note that R&D is not covered under this scheme.)
- ISO 9003: quality assurance in final inspection and test[13]. Conformance by a supplier only at the final inspection and test.

ISO 9001 and ISO 9000-3 compared

In the case of software, ISO 9001 is not specific enough and this has resulted in the production of ISO 9000-3 guidelines specifically for software[14]. The reason for this is that with software there is a need to coordinate the activities of both the purchaser and the supplier to ensure that the delivered product is fit for purpose. More detail of the ISO 9000-3 Quality System is presented in Table 1. This information is taken from

ISO 9000-3[14] and will be interpreted by each individual organization that implements the guideline.

In particular, in the case of software, in contrast to normal R&D activities, both the supplier and the purchaser have responsibilities for the specification, selection, installation and support of the software product.

The first of the user's responsibilities was outlined in the previous V&V column on the URS[1]. Of course, if the purchaser ignores their responsibilities then one of the main principles of ISO 9000-3 simply collapses. This is the quickest way to throw the investment in a software product down the drain. In addition, consider the lingering death of the system you will purchase with no URS, and the frustration you will have explaining to the boss why your system does not work.

The uniqueness of software is shown in the fact that the ISO 9000-3 guideline is double the size of the ISO 9001 document; a simple, but empirical, method of demonstrating the complexity of the system development life cycle. Furthermore, if you read both documents, there are some differences in their phraseology: ISO 9001 uses in all sections the word 'shall'; however, in ISO 9000-3 there is a mixture of 'shall' and 'should'. The basis of an interpretation for ISO 9000-3 is that where 'shall' is used the vendor must perform the activity, whilst 'should' is a recommendation but is not mandatory[10]. This allows scope for the vendor to interpret the guidelines according to working practices and the products they are developing.

Kehoe and Jarvis[10] also present an interesting critique of ISO 9000-3; the three main ones are presented below:

- The word 'quality' is overused and blurs the boundary between the engineering aspects and the quality-assurance activities necessary to build quality into the product. This, therefore, makes it difficult to define the roles and responsibilities within an organization between the functions responsible for designing, building and testing the product with those responsible for ensuring that the quality has been built into the product.
- The requirements phase of a software product is missing. The guide moves from the user's requirements to a design of the product. The truth is often different as companies will write a marketing requirements specification before designing the product.
- The need for a quality plan mixes the topics that are actually part of the engineering phase of a software project. For example, "defined inputs and outputs from each development phase of the project," is normal software development practice.

The ability to interpret the ISO guidelines has led, in the UK, to the development of the TickIT software guidelines, which are still based upon ISO principles but with levels of quality for software[15]. Auditing TickIT and non-TickIT vendors you can see a difference, which gives a better degree of confidence in the procedures of the former vendors. However, still remember that this does not guarantee product quality of fitness for your purpose.

Therefore, look closely at the ISO certificate from your vendor. Which version of ISO is it? What is the scope of certification? Would a computer application produced by a vendor with ISO 9002 certification be useful for you? Consider the design: it is not covered under this scope of certification. Why? Was the application designed on the backs of used envelopes and cigarette packets? Alternatively, with an increasing number of companies, the software may be written in one country and the marketing role has the ISO certification. Again the bottom line is do not take things at face value.

ISO 9000 and regulatory compliance

In The Gold Sheet[16], there is discussion on why ISO 9000 does not assure GMP compliance. The main problem is the need to develop a reliable third-party certification scheme for software houses that regulatory authorities will recognize. Ken Chapman, for many years chair of the Pharmaceutical Manufacturer's Association (PMA) computer validation committee, states that ISO 9000, "won't do a thing for structural integrity (of software)[16]". Chapman has been involved with many audits and the failure rates of both ISO-certified and non-accredited organizations were both high. Furthermore, ISO accreditation does not guarantee a 100% problem-free product or service. Remember how the quality manual should have a customer complaints section. At least you can be assured that your complaint will be documented, but not cleared within a specified time unless that is in the procedures.

Furthermore, ISO is a voluntary scheme. You voluntarily enter and you can voluntarily leave the scheme. Following this, if there are problems, the accreditation body can suspend an organization's certification, or the organization can voluntarily withdraw from the scheme.

With programming costs rising in the developed world, many companies are looking to outsource their programming to Third World countries where these costs are lower. Some companies that do this can be ISO 9000 registered; however, often the scope of accreditation means that only the overall process may be covered. The detailed programming and testing may have been done in another country, usually India, which writes about 50% of the world's software.

Marketing hype and contracts

It is interesting to look at the Dr Jekyll and Mr Hyde approach of companies to marketing their products through proposals and protecting themselves through contracts. For large purchases, if you have not already done so, it is very useful to read both the marketing material and the contract. This can be very instructive as we can see from the sales response to questions on compliance with GLP Guidelines and their contact terms and conditions.

Table 2 is an actual example of a proposal from a company with ISO 9001 and TickIT certification. In the left column we can see the proposal showing the benefits of the quality approach and the way that regulatory inspection can be facilitated. Compare this with the right column which shows the appropriate section from the same contract stating that the company is not responsible for errors in its own software.

This is an interesting contrast. Vendors are performing a delicate balancing act between trying to sell the product whilst maintaining their legal position. The difference between the two is the type of person who is writing, the individual document and their aims. Again, the bottom line is to balance the sales message with the realism of the contract.

This also reinforces the fact that you, and you alone, are responsible for ensuring the product you purchase is fit for purpose.

Area	Detail
Quality system framework	Management responsibility
	Quality system
	Internal quality system audits
	Corrective actions
Life cycle activities	General
	Contract review
	Purchaser's requirements specification
	Development planning General, Development plan, Progress control, Input to development phases, Output from development phases, Verification of each phase
	Quality planning
	Design and implementation General, Design, Implementation, Reviews
	Testing and validation General, Test planning, Testing, Validation, Field testing
	Acceptance (Testing)
	Replication, delivery and installation
	Maintenance
Supporting activities	Configuration management
	Document control
	Quality records
	Measurement
	Rules, practices and conventions
	Tools and techniques
	Purchasing
	Included software product
	Training

table 1 Overview of the scope of ISO 9000-3.

The ISO 9000 bottom line

Do not take ISO certification of any organization at face value: the type and scope of accreditation could be crucial to your decision to purchase. Let's look at some examples from companies that have ISO 9001 certification for their scientific products:

- An instrument's software that has the facility for the user to delete a data point they don't like simply by double clicking on the point and confirming the deletion.
- A data-capture instrument and associated software in which the user password is stored unencrypted in the .INI file, which can be easily accessed by anyone with basic knowledge of Windows.
- An application in which, under certain circumstances, the data displayed on the screen can be different from the print out.

The impact of these features has to be assessed under the umbrella of their likely use. In a regulated environment these features would be unacceptable in contrast with a research environment in which the impact would be inconvenient, but with less impact.

Again, there is nothing in ISO 9000 about fitness for YOUR purpose, nor the quality of the product. These will always remain the buyer's responsibility — as the regulations, guidelines and vendor's contracts make absolutely clear. Being informed is better than being unaware, and you can make better decisions. ISO 9000 or not, buyer beware!

What if the supplier does not have ISO certification?

Having spent some time discussing ISO 9000, what about those vendors that do not have an ISO certificate? Some may have written quality systems that they follow but have not acquired accreditation because of cost or other reasons. Other companies may have nothing. The key is what is their attitude to quality? If there is no regard for quality or improvement, the answer is very simple. Walk away. Do not consider these companies.

If the product is right but there is no quality system but a willingness to work towards an overall quality approach, then you have a choice that essentially balances risk. Some of the risks are:

- what does the product do?
- how critical is it?
- what happens if the company goes out of business?

It is difficult to give specific advice as this requires a detailed knowledge of the situation. We will look at this area again under vendor management.

In part II of this article Bob McDowall will discuss the full scope of an audit, the role of a checklist, how to conduct a successful audit and vendor management. A complete set of references will be provided at the end of part II.

The Proposal says	"Software products are designed to operate with standard computer hardware in the typical laboratory environment. Their design functionally integrates laboratory instruments and results in a unified information architecture that allows for more effective use and control of laboratory data. This enhances the value and utility of the data, especially when measured against regulatory compliance. Validating laboratory software is becoming more complex resulting in escalating costs and longer timescales. Because the company has achieved ISO 9001 and TickIT certification, while stressing a high level of commitment to open industry standards and software design, customer validation is made easier. ISO 9001 with TickIT certification can significantly reduce time on customer audits of vendor facilities as well as user functional validation."
The Contract says	"The company makes no warranties that errors have been completely eliminated from any licensed software. The company makes no other warranties, expressed or implied, including but not limited to fitness for a particular purpose or merchantability with respect to any licensed software."

table 2 Comparison of the Proposal and Contract of an ISO 9001 and TickIT certified software vendor.