



How to Manage the IT Helpdesk

A guide for user support
and call centre managers

Second edition

Noel Bruton

2002

BUTTERWORTH
HEINEMANN

OXFORD AMSTERDAM BOSTON LONDON NEW YORK PARIS
SAN DIEGO SAN FRANCISCO SINGAPORE SYDNEY TOKYO

The need for control is a point already made in the previous chapter – essentially that if you do not control the workload, it controls you, to the benefit of nobody. Measurement provides an indication of what is happening; this chapter looks at the issues of controlling and managing workflow and considers some of your options.

23.1 Procedures and standards

As in other parts of industry, formal procedures and standards for helpdesks and technical support offices have become increasingly fashionable in the last few years. One notable influence on the movement is ISO 9000. Under this standard, formal procedures are required to be fully documented and accompanied by a *quality manual*. Evidence must be produced from time to time that these procedures are being adhered to, otherwise the company risks the loss of its accreditation of compliance with the standard. Another standard, aimed squarely at IT, is in use at an increasing number of IT support desks in the UK Civil Service as well as non-governmental and private sectors. These are the methods described in a set of manuals known as the *Information Technology Infrastructure Library* (ITIL), published by the UK government's Central Computer and Telecommunications Agency (CCTA). ITIL has been taken a stage further, by becoming the subject of training programmes leading to academic qualifications, which are themselves recognized by some companies.

There is no doubt that much benefit is to be gained from formalizing procedures, and ISO 9000 offers a way to measure standards of management regardless of industry. Compliance with ISO 9000 purports to be an indicator of professionalism and completeness in the way the business is run, and this can imply that the work done by a holder of ISO 9000 accreditation is of high quality. However, the standards advisers do not tell a company how to do its job; it cannot, the standard is too general and applicable to too wide a range of industries. ISO 9000 offers an opportunity to set in stone the current standards of your work and document them to ensure strict compliance. Ostensibly, this ensures consistency of work; and if a high quality were documented at the outset, then compliance to the documented and accredited procedures would apparently ensure consistent high quality. That means, however, that the opposite is also true – that if the documented procedures oversaw a low quality of work, then what would be produced would be a consistently low quality. At the time of writing, a new version of ISO 9000 is gaining ground, which deals admirably with this criticism by implementing in a standard a process of continuous improvement – thus compliance with the standard will now imply an improvement in the quality of the product or service.

ITIL differs from this in that it purports to show support and other IT managers how to do their job; it describes techniques and methods. Personally, I have some strong differences with ITIL; I find it inward looking, uncommunicative, not customer-oriented, and too focused on technical issues to be considered a complete operational guide for a helpdesk or IT support service environment. ITIL does not do enough to inculcate into the helpdesk the fact that there are real people out there among the user population. The result is that it worries too much about staffing and knowledge, and not enough about user expectations, communication and motivation. Do not misunderstand my views on ITIL – it has much to recommend it. On matters of internal organization and workflow control in a user support department, it is unsurpassed. Its volumes *Helpdesk* and *Problem Management* are recommended reading for all support managers. However, I have stated often in this book that I feel these issues are at best of equal importance to the marketing of the helpdesk. ITIL has in the past been decidedly weak on anything to do with client management, and I see this weakness to be fundamental. However, ITIL too is in good hands – and reviews of its methodologies and philosophies are constantly taking place.

Having read my thoughts, should you ever have cause to study ITIL as well, you may find some of my reasoning at odds with much of theirs; that is good, you will have the opportunity to make up your own mind.

It is this opportunity to make up your own mind which leads me to my conclusion on external standards such as ISO 9000 and ITIL: that they can never be absolutely right for anybody. The firmness with which ISO 9000 sets procedures has trapped some compliant firms; they have not allowed for enough flexibility in their procedures, so they have restricted their options for change in a changing world. The dilemma of some ISO 9000-compliant firms is either to remain compliant, thus possibly missing market opportunities by being unable to change fast, or to risk losing accreditation by failing to remain compliant, thus becoming less attractive to those clients for whom ISO 9000 compliance is a main requirement in a supplier. Either way it is clearly a loss of flexibility and the only way to deal with it is to build into the formal procedures the anticipation of change. Change management procedures are notoriously difficult to document; after all, ISO 9000 strives to maintain consistency, and so I find it inherently at odds with change. If I may be momentarily flippant, I have yet to see in an ISO 9000 procedures manual the procedure 'In the event of rapid and unforeseen market change, this procedures manual shall be consigned immediately to a swiftly forgotten pit in a distant corner of the company car park' because that would probably be judged to be a non-compliant procedure.

In the absence of an external demand for you to formulate and abide by procedural standards ordained by an external authority, the best procedures for you and your own customers will always be the ones you design for yourself. By all means take advice in the design of your procedures, but in the end they should be yours. If they are not yours, the risk is that you will find them difficult to comply with, and very difficult to sell to your staff. Computer technicians are notorious for their resolve to retain their freedom of action. And if your procedures and standards look to them like unnecessary imposition – then somebody else's procedures may look like an occupying foreign army. Imposition incites resistance.

25.1 Acquiring knowledge

One of the most obvious ways to acquire technical knowledge is to go on a training course. This is all very well for new knowledge, but it is seldom the only way or even the best way for increasing the knowledge of technicians in areas they are already aware of. Take the example of the computer technician who knows the spreadsheet package Lotus 123 very well and now needs to be able to support a rival spreadsheet, Microsoft Excel. It would be inappropriate to send this technician on a 'Microsoft Excel for Beginners' course, as some of the course content would probably be on the principles of spreadsheeting, an area the technician already understands well. Boredom would result, possibly along with unconstructive comparisons of the products.

For technicians, training courses are rarely the best way to learn about new products; they would learn more from playing with the products themselves. Technicians benefit from training courses where what is being taught is a new technology rather than a new product.

'Playing with the product' is a euphemism for a more rigorous process the learning technician should go through with a new product. 'Play' is non-directional – the technician needs something more directed, so the process should have an objective. Examples of learning objectives are 'to expose the technician to the product from the user's point of view' or 'to increase the technician's spot rate (or fix rate) on this product by 10 per cent within one month', or 'to report on areas of the product where user difficulty is anticipated'. This way the technician learns the product, but with the direct and measurable result of an improvement in service.

25.2 Where should the knowledge be?

If you operate some kind of helpdesk service, where the users telephone their problems in, the issue of where the knowledge

should be in user support is a key one. The fact that the users have chosen to use and presumably you have chosen to offer such an immediate communications method as the telephone suggests they want the answer now. Even if the actual problem may not be urgent, wanting an immediate answer is quite natural – after all, it is cheaper and easier from the user's point of view to request the knowledge exactly when she needs it, rather than have to wait for the knowledge to arrive, or learn it in advance and wait for an opportunity to use it. The use of a helpdesk to get knowledge precisely and only when you need it is just another form of 'just-in-time' deliveries.

When the users call the helpdesk, they want the information now. Because it will disappoint the users to have to escalate the query, because it will depress your spot rate, and because it costs more to escalate, the helpdesk will endeavour to have the requisite knowledge available at the helpdesk.

Some companies see the knowledgeable helpdesk as the high-quality service option; some see it as an opportunity to improve user productivity; others see it as a way of saving money, as the more knowledge the helpdesk has, the fewer resolvers are needed.

Many products exist to help the helpdesk become more knowledgeable, and most of them come in a form of knowledge tools. Often these tools are built into commercial helpdesk packages, and some even go so far as to feature an ability to store scanned pictures and graphical illustrations of potentially useful items of knowledge.

Some companies see these knowledge tools as an opportunity to put in place a helpdesk which is to all intents and purposes technically ignorant; the databases are expected to provide the knowledge the caller or user requires and all the helpdesk does is pass the information on. I find there is only a limited range of helpdesks where such a proposition would be entirely practical; many if not most requests to IT helpdesks are either too simple ('my terminal's locked, would you recycle it please?') or too complex ('how would you recommend I go about X?') to be solved just by a database, no matter how comprehensive, and the patience of the caller would be severely tried by such a process.

There are probably two main types of these tools – the first is the knowledge database, which stores text and graphics in discrete documents of varying size and complexity. By querying this database, a document can be found to give an answer to a given

technical problem. The query is normally entered as a series of keywords ANDed, ORed or NORed together.

The beauty of a database such as this is that you can put virtually anything in it; in fact one is encouraged to put everything in it, as this will increase the likelihood of a match to an enquiry. Often, database entries are produced as a result of a user asking a question and the technician researching and documenting the answer. The database can become the repository of all the technical knowledge about old products, when you would rather the technicians cleared their heads of old knowledge to make way for knowledge about new products.

But in the strength of such databases also lies their weakness. The scope and flexibility of them can lead them to reach unwieldy proportions. A query can produce too many answers, so the mere list of 'possibly relevant' document titles can be unusably large. Or the query may produce no answers at all because the database is unaware of that product. Either way the credibility of the database is undermined. As an experiment, I once linked an 18,000-document database direct to the call-logging system of a particularly busy helpdesk. Every time the database was used, a flag was set that I could later count. I found that the database was consulted for around 2 per cent of the queries and useful for even fewer; so the database was virtually useless, despite the fact that I had deliberately invested in its content by having staff and systems dedicated to filling it with relevant documents.

The second major type of knowledge tool is the decision-tree type. This is a structured device which asks questions to try to arrive at an answer by a process of elimination. Even more sophisticated types of such trees are appearing, imbued by artificial intelligence with a rudimentary diagnostic ability. The application for tools such as these is in the ignorant helpdesk. The range of questions may be so wide that it is neither justifiable nor practical to train the helpdesk staff to be able to answer the questions – so either they or the enquirer consult the decision tree.

These too have their problems; for example, the helpdesk staff may learn faster than the machine does and eventually cease to use it for certain types of problems, raising the risk of reduced accuracy of answers; the correct level of simplicity of decisions may be difficult to arrive at, so the enquirer must exasperatingly go through the same questions every time before arriving at something new; a wrong response to the machine's question is given at some point, leading the enquirer up a blind alley.

How to Manage the IT Helpdesk

For both these knowledge devices, some problems will always remain. One is that the system has to be administrated. There has to be somebody there to fill it up with the right information, ensure it is up to date and in a usable form. Often, helpdesks turning to a tool such as this underestimate the secondary investment needed to manage the system and eventually it falls into costly disuse simply because they never allowed for the resource needed to feed the machine.

Another problem is in the use of the tool. When a user calls the helpdesk, he or she is unlikely to want to wait while the helpdesk operative scrolls through a list of possible solutions or navigates round a decision tree, especially if the payoff does not appear to come quickly.

Knowledge is fashionable – the types of questions posed to the helpdesk will change rapidly as the user population acquires its own knowledge by experience and ceases to ask certain questions; at the same time, new products come and go all the time, changing the type of knowledge we need. Any knowledge tool must take account of this simple truth before it can be truly effective. As things stand, I rarely find examples of where knowledge tools could really be as much of a benefit to a helpdesk and its customers as could a proper culture of personal knowledge acquisition.

Look through the job advertisements of any computer journal read by IT support people for an indicator of what most corporations seem to regard as the key attribute of helpdesk and support staff. The need, desire, hunt for technical knowledge screams at you from the pages. The ads read like a list of computer software products, and the closest match of the candidate to that list is the first step towards being interviewed.

The software and hardware manufacturers also appear to lay great store by this. Novell, for example, has a range of qualifications which can be pursued by computer technicians. These qualifications have become a marketable commodity: a technician whose technical competence is acknowledged by a Novell-endorsed certificate commands a higher salary and is in greater demand than one whose ability is not so endorsed. A whole market has sprung up to provide Novell-recognized training courses, with a range of quality, pace and price to suit every pocket.

Cynics like me would be very suspicious about training programmes such as these. Of course they create some kind of measure of a technician's knowledge, and that is a useful aid for the company seeking to employ a proven network technician. But a certificate is rarely a true indicator of real knowledge, it is more an indicator of the technician's ability to get through a training programme. The knowledge itself is transient; in six months, Novell will probably have another version of their operating system, in two years it may well look very different

than it does now; but the certification will still be valid, even though what it once measured is no longer current.

With these certification programmes, software companies have nothing to lose and so very much to gain; they can sell training courses, appoint recognized trainers, perhaps for a fee. They can ride on the back of the hype the qualification brings in its wake. They can reduce their support burden by encouraging their customers to pay to be able to do their own support. For all the benefits this training brings to the world in general, you can bet your boots it brings a whole lot more to the company that endorses it.

However, I think this relentless pursuit of technical acclaim distracts so much from the attributes we really want in our technicians; and the attributes we seem to mention in our job advertisements all too rarely.

If our technicians only ever dealt with machines, then technical ability is all we could ever reasonably ask of them. But they do not – they deal with people. The true helpdesk and user supporter has to deal with people for the function to have any meaning. Let us go back to the point: the computers were put here to increase the productivity of the white-collar functions of the company. If that increase in productivity cannot be assured, then the investment in computer technology is wasted – worse, it does more damage than good to the corporation's finances.

The only IT function which is solely dedicated to maintaining continued, hour-by-hour productivity through IT usage is computer user support. Therefore, what we need in that department are people and methods that increase or maintain user productivity; and user productivity is not just an 'I've got a bug in my software' issue, it is a 'Help me to know how I go about producing the sort of output I need' issue. The first query requires technical ability alone; the second query is in my experience if not more common then more demanding, and requires technical knowledge coupled with an ability to convey that knowledge in terms the user can understand and believe in.

Without the ability to communicate technical knowledge in stressful and challenging situations, then all that expensively acquired technical knowledge is worth a great deal less than you paid for it. I will go further; it may mean that you have wasted your investment because the knowledge can only ever be used on machines, never on people, so it can never be exploited to increase user (and therefore corporate) productivity. Put simply,

technical knowledge without communicative ability costs a lot, but realizes comparatively little benefit. In my opinion, the common tendency to seek computer technicians for their technical ability alone is at best extravagant, at worst futile.

So why do we seldom ask for communicative ability in our job advertisements? One reason may be that it is too subjective and thus difficult to measure; communication is largely a question of rapport, and a good rapport between interview candidate and interviewer does not guarantee a continued good rapport with the users if that candidate is hired. Another reason may be that companies looking for technicians feel that the technical knowledge is the basis for considering a candidate and that the personality can be assessed during the interview.

I too have made the mistake of hiring a technician because I desperately needed his technical knowledge. He cost me a disproportionate salary and was with the company less than a year. During his brief stay with us, although he gained the respect of other computer engineers, he never did gain the respect of the users he was hired to serve. My mistake, and one I hear echoes of from other helpdesk managers. The lesson I learned was that if I need technical knowledge in a hurry, then call in a short-term contractor and milk their knowledge by having them train your own staff; and after they have gone, kick yourself for not having seen the need for that knowledge coming, so next time, plan better.

In my training courses, I ask helpdesk managers the question 'What are the attributes and skills of the ideal support person?', and write the suggestions on a flipchart as they are called out from the floor. Typical responses are as follows:

- *Patience* – to be able to listen to a user describe a situation you have encountered many times before yet still be able to patiently explain the solution; and to painstakingly go through the diagnostic process in search of the cause of a problem.
- *Assertiveness* – to give the user confidence in your ability to solve the problem; to be able to deal with a user whose expectations are unrealistic; to be able to treat the helpdesk manager as an equal so as to recruit him or her as a resource.
- *Thoroughness* – to make sure the job is complete, that the problem is truly solved, the user satisfied and the paperwork filled out.

- *Enthusiasm* – to enjoy the job and stay motivated; to remain positive in a challenging situation; sometimes, to convey cheerfulness to a user who might otherwise think that a situation is irredeemable.
- *Responsibility* – to be able to take on the burden of a task, set oneself an objective and maturely follow it through to a successful conclusion without unnecessarily involving others.
- *Technical knowledge* – to have acquired, and continue to acquire naturally, the sort of technical knowledge the job requires.
- *Empathy* – to be able to put oneself in the user's position so as to understand the real nature of the difficulties the computer presents them with, and therefore to treat the problem with the appropriate priority.
- *Communicative ability* – to be able to use language well enough to convey confidence, to ask the right sort of questions to solicit information about the nature of the user's problem, to use the right register and phraseology so that the user can understand the solution as described without being alienated by technical jargon. This skill is as important over the telephone as it is face to face, although the challenges of the two situations differ considerably.
- *Works well under pressure* – this can range all the way from staying amusing and positive when the office gets noisy to being able to handle quite dangerous levels of stress.

When the ideas are exhausted, I then go round the room and ask at what age people typically begin to acquire their technical computer knowledge; the responses range from mid-teens to late twenties. I follow this up with a question about what age people acquire their ability to communicate, their patience, empathy and enthusiasm; these are much earlier, with most respondents acknowledging that these are personal skills and largely acquired in childhood, maturing in later years.

The fact is that technical knowledge is learned by experimentation and education, and relatively late in our personal development. We can continue to acquire technical skills throughout our lives, and from a corporation's point of view that means that where they are missing, they can be taught. It is also possible to change and improve personal skills, where they are lacking – but this is much more difficult.

What this all comes down to is that the ideal user support technician needs to have the appropriate personal skills to deal with the users – for the users come first. What our ideal helpdesk

needs is genuinely personable people who, by the way, are also accomplished computer technicians. Indeed why should the corporation ever take on the burden of changing dysfunctional or inappropriate personalities? It is much easier to hire personable people and teach them to be better technicians than it is to hire clever technicians and teach them to be better people.

27.1 Inherited staff

However, this ideal of hiring good people and developing them technically is in practice one that can only really be pursued during recruitment. When staff are inherited rather than chosen from an open field, then it is more a case of making the best possible use of what skills and attitudes are available, while improving in areas of weakness.

It is difficult, but not impossible, to change and improve the non-technical skills of our staff. The first hurdle we have to get over is the attitude held by some organizations that personal and social skills are a prerequisite for employment, and thus the corporation should take no responsibility for helping people who fall short in that area. This attitude is justifiable to some extent – but it can also be self-defeating. If the corporation gives no clear indication that good human interaction is desirable, while simultaneously indicating (as many do) that technical skills are vital, then technical employees can hardly be blamed for following that emphasis by conducting themselves and their services as though computers were more important than users. The most effective way to get technical staff to engineer good user relationships is for the corporation's managers to publicly lay as much store by, say, good communications skills as by the ability to configure a network server. In practical terms, that means using the same management techniques for personal skills as we do for technical skills. That means advertising for good personal skills, not just for recruitment purposes but also because your own staff will see those ads and so will see what it is you are looking for. It means sending people on personal development training courses as well as teaching them to be better computer engineers. And it means identifying and measuring the growth of social as well as technical prowess.