Why Did That Happen?
A Brief Explanation for the Proliferation of Barely Usable Software in Healthcare Systems

C.W. Johnson,

Department of Computing Science, University of Glasgow,
Glasgow, Scotland, G12 9QQ.
http://www.dcs.gla.ac.uk/~johnson, Email: johnson@dcs.gla.ac.uk

Abstract

Clinicians and support staff are faced with increasingly complex computer applications. This complexity stems, in part, from the integration of heterogeneous systems ranging from computerized patient records to theatre management and dosage planning applications. Complexity also stems from the increased functionality offered by this new generation of IT systems. Many members of clinical staff are bewildered by the vast array of configuration options and operating modes that are supported by computer-based systems. Conversely, manufacturers often feel compelled to offer more and more software features in order to retain market position. These factors combine to create 'usability' problems that have had a direct impact on patient outcomes. The poor usability of clinical software also has a number of indirect effects. For example, the costs of replacing and upgrading inadequate computer systems carry significant opportunity costs in terms of the services that might otherwise have been funded.

This paper addresses the following issues:
1. There is a tradition of 'making do' with poorly designed software that should be questioned.
2. Poor 'usability' has a direct impact on patient outcomes;
3. Poor 'usability' also has an indirect impact on opportunity costs of replacing computer systems;

In the future:
1. We need to educate staff to reject substandard computer interfaces early in the acquisition process;
2. We need to encourage the use of human computer interaction techniques in healthcare;
3. We need to train staff to recognize the dangers of 'working around' poor interface design.

Introduction

'Usability' is a difficult concept. It describes the users' ability to access and operate the functionality provided by complex systems. However, designers must consider a myriad of issues in order to achieve this goal. Users must observe necessary information displayed on output devices. It must also be physically possible to provide appropriate input. This can be non-trivial, for example when surgeons or anesthetists must access information during surgical procedures. Usability also implies the need to match interaction with the cognitive and perceptual capacity of the user in their working environment. For example, an additional auditory warning is unlikely to attract the user's attention if they are already surrounded by multiple concurrent alarms. Similarly, it can be difficult for users to learn to operate systems that exploit terms, concepts and language that has little meaning for them. In particular, many healthcare systems seem to assume that clinicians will have a detailed understanding of computer networks and architectures [1].

For more than twenty years, the fields of Ergonomics, Human Computer Interaction and Human Factors have developed tools and techniques to improve the usability of complex systems. For instance, participatory design encourages the consultation of end users throughout the development process. A representative of the end-user community will, typically, be recruited to join the development team for the lifetime of the project. This approach is not as simple as it might appear; developers often have to balance the competing needs of different users groups. The members of one department can oppose the views of their colleagues in another. Participatory design techniques provide means of identifying and resolving such potential conflicts early in the development process. Other tools can be used in a more direct manner to ensure that designers choose appropriate fonts and restrict their use of color to reduce the perceptual problems that affect many computer-based systems.

Relatively little attention has been paid to the usability of software systems in the healthcare domain. Techniques such as participatory design are seldom used. This is regrettable because computer-based applications now support everything from theatre management through to the programming of infusion pumps. Device complexity contributes to the usability problems of clinical software. It can be difficult for end-users to find the time to learn about the subtle differences between many different operating modes and functions. It can also be difficult to review the thousands of calibration and configuration procedures that are supported by complex computer-based dialogues spread across many different forms and menus. These usability problems are exacerbated by the lack of well-structured documentation and manuals. Conversely, manufacturers often feel compelled to offer more and more software features in order to retain market position. The use of intermediate medical equipment suppliers can also make it difficult for manufacturers to contact their users directly in response to technical queries or to provide product updates that might address potential usability problems.

Opportunity Cost and the Indirect Impact of Usability on Computer-Based Systems

A number of failures have affected the acquisition and deployment of computer based systems in the UK National Health Service (NHS). These include the Wessex Regional Information System Plan, the Hospital Information Support Systems Initiative and the Clinical Coding Information System [1]. These failures have complex causes, including managerial problems and technical difficulties. However, they all involve a failure to consider end-user requirements early in the development cycle. Arguably the most notorious of these incidents involved the London Ambulance Computer Aided Dispatch system. This was intended to provide ambulance drivers with computer-aided support in locating their destinations as they drove around the UK capitol [2]. As the demands on the system rose, information about the location of each ambulance became increasingly out of date. This created error messages that were presented to the drivers. The increasing number of these warnings added to the users' frustration with the software. As a result, drivers became less and less inclined to update essential location and status information. This, in turn, led to more error messages and a vicious cycle developed: "The situation was made worse as unrectified exception messages generated more exception messages. With the increasing number of "awaiting attention" and exception messages it became increasingly easy to fail to attend to messages that had scrolled off the top of the screen." ([2], Paragraph 4023)

The costs of individual health system failures are significant but not startling by comparison with other computer-related projects. The London Ambulance Computer Aided Dispatch System was estimated to cost in the region of $2.2 million and was never fully operational. In contrast, the London Stock Exchange paid almost $130 million for the Taurus trading system. Securities companies paid a further $600 million even though no modules were implemented [3]. It is important, however, not to underestimate the cumulative costs of these failures on cash-limited national health systems [4]. Research studies in other areas have shown that customers regard over 20% of their expenditure on information systems as 'wasted' [5]. There is evidence to believe that this percentage is even higher within the UK NHS and the US healthcare system [1]. This is because many service providers cannot access true economies of scale through software reuse. Even in the UK national system, centralized management cannot demand that every hospital or GP surgery install a uniform set of IT systems. In consequence, every trust or general practice can have its own IT strategy. This has created a mixture of legacy systems and piecemeal acquisitions. Given this background and the lack of specialist software project management expertise, it is hardly surprising that so many projects fail and that those systems which do succeed are often barely usable [2,3]. A number of initiatives have been set up to address these problems. The UK National Health Service Steering Group on Health Services Information was set up in 1980. This body helped to coordinate the provision of computer-based systems across UK healthcare. However, subsequent initiatives including *Getting Better with Information* and *Information for Health* and *A strategic framework for public services in the Information Age* have focused on the development of new software applications, the wider provision of terminals and the extension of network services. Very little attention has been paid to the problems of developing computer-based systems that can actually be used by its staff.

The US healthcare system is very different from the UK NHS model. There is, however, ample evidence that a similar failure rate affects computer-related projects and that many of these problems stem from usability issues [6]. Although the US system is less centralised, government funded projects face many of the problems described within the UK NHS. For instance, the US Department of Veterans Affairs and the

Department of Defense have a combined budget of some $34 billion for the provision of healthcare services. However, they maintain patient information in separate systems. In December 1992, Congress proposed the use of an integrated information technology application to provide 'greater continuity of care…and save software development costs' [7]. The intention was to deploy the Government Computer Patient Record (GCPR) system on October 1, 2000. However, target dates were not met and cost estimates were unreliable. In September 1999, GCPR was estimated to cost about $270 million over its 10-year life cycle. This had risen to $360 million by August 2000. By the end of 2000, the US General Accounting Office found that 'in the near term, physicians and other health care professionals would not have access to comprehensive beneficiary health information across the agencies, limiting the extent to which the effort will provide the benefits originally envisioned—including improved research and quality of care as well as clinical and administrative efficiencies'. An interim system was designed but this suffered from major limitations. For example, physicians at Military Treatment Facilities will not be able to view Veteran's health information or information from other Military Treatment Facilities. Requested data can take as long as 48 hours to be received. In consequence, the General Accounting Office questioned both the usefulness of such shared information and the overall usability of the interim system.

Usability problems are not confined to large government IT projects. They also affect a host of other health-related systems. For instance, the FDA's analysis of 3,140 medical device recalls conducted between 1992 and 1998 reveals that 242 (7.7%) were attributable to software failures [8]. Subsequent studies have found that this proportion has risen sharply since this original study [1]. Of the software related recalls in the initial FDA study, cited above, 192 (79%) were caused by software defects that were introduced when changes were made to the software after its initial production and distribution [8]. The majority of these updates stemmed from 'usability' problems. The FDA concluded that 'software validation and other related good software engineering practices … are a principal means of avoiding such defects and resultant recalls' [8]. These findings led on to the development of best practice guidelines that are intended to ensure that usability of complex medical devices, including software related systems.

> "If device operation is overly complex or counter-intuitive, safe and efficient use of a medical product can be compromised… The application of user interface design principles and participation of healthcare practitioners in design analyses and tests are very important. In addition to increased safety, an added benefit of such practices is the likelihood that good user interface design will reduce training costs to healthcare facilities". [9]

The final sentence in this quotation from the FDA guidance document is very important. Rather than looking at the costs of usability problems, it is important also to look at the potential benefits of improving end-user interaction with healthcare software. For instance, Karat's cost-benefit analysis cites projects where spending $60,000 on usability engineering throughout development resulted in savings of $6,000,000 in the first year of operation [10]. She argues that greater savings can be achieved if the same organisation both develops and uses the software application. The overheads associated with rewriting the software in response to user complaints can be reduced. There are also savings in terms of employees' time in working around any initial problems with the system. It is possible to question the basis of such assertions. They are often derived from extrapolations based on a small number of high-profile projects. However, these arguments have had a powerful effect in motivating initiatives such as the *usability.gov* site run by the US National Cancer Institute. This provides a useful starting point for readers who are interested in the underlying practices and principles of usability engineering.

Coping Strategies to Combat Software Usability Problems
Previous paragraphs have argued that usability problems impose significant financial burdens on healthcare systems in the UK and in the US. The following pages go on to look in more detail at the nature of these problems and the way in which personnel respond to them. For instance, clinicians and technicians have developed a range of coping strategies to overcome the poor design of many user interfaces. These strategies include a reliance on local experts who themselves may have only a rudimentary grasp of the software they are using. Coping strategies also include the development of unofficial 'local' manuals that replace those published by the manufacturer. These can suffer from omissions and errors that undermine the safety of many applications. For instance, a recent study of an adult Intensive Care Unit observed that portable monitors were being used when patients were transferred between wards [11]. On one occasion,

the monitor switched itself off with a "BATT COND" warning even though there appeared to be sufficient power to drive the device. The user manual revealed that the battery must be replaced after the fiftieth time it is used even if there is sufficient charge for the monitor to operate. Unfortunately, there was no way for clinical staff to determine how often a battery had been used. This represents a design failure. The device must record the number of times it has been used in order to trigger the 'BATT COND' warning. It should have been a relatively trivial matter to present this information to nursing staff, for instance through the patient monitor display. Staff lacked the time necessary to record each time they used a particular battery pack and so they eventually resorted to a more ad hoc coping strategy. Quickly removing and replacing the battery during patient transfers could suspend the device warning. Such 'work arounds' typify many interactions with clinical systems [1]. However, they carry a number of risks. For example, the limit of fifty operations is based upon the manufacturer's predicted safe life for the battery. It is likely that at some point a battery will have insufficient power to drive the monitor after being reinserted by nursing staff. Similarly, there is a danger that over time staff may 'trick' the device into going well beyond fifty operations. There are economic pressures to extend the life of many devices. They might also be tempted to continue 'tricking' the device if replacement batteries are in short supply. The incentives to exploit these coping strategies only increase if staff members feel that the 'fifty use' limit is too conservative a constraint on the everyday use of the device. Such coping strategies need not directly threaten the safety of the patient being moved. However, previous adverse events have shown that such behaviors can combine with other monitoring failures with far more serious consequences [1].

Software Complexity and the Limitations of Coping Strategies
The impact of usability problems on the operators of healthcare systems can be seen in many incident reports. For example, the following account is taken from the Manufacturer and User Facility Device Experience database (MAUDE). The Centre for Devices and Radiological Health (CDRH) within the US FDA is responsible for maintaining this system. MAUDE is updated every quarter with voluntary reports of adverse events involving medical devices. The following account focuses on a centralised patient monitoring system. It shows that the complexity of many applications can prevent users from diagnosing potential problems. This reduces the risks associated with inappropriate coping strategies because it can be difficult to identify 'work arounds' if users cannot explain the problems they observe. Such 'gains' are, however, outweighed by the usability problems that arise when healthcare software suffers apparently random failures. In this incident, the data associated with one patient would be appended to the record of a different patient if the first patient were moved from one monitoring point to another in the hospital. However, clinicians had great difficulty in recreating the conditions in which this failure occurred. The data would only be incorrectly appended to the record of a different patient if the first patient were entered in 'AUTOADMIT' mode. The problem would not happen if 'MANUAL ADMIT' had been used. The problem did not occur if the first patient was returned to the same monitoring point, for example after treatment elsewhere in the hospital, if no new patient had been entered for that point in the meantime. None of this affected the real-time monitoring alarm system. Even once the company had identified the context in which the incident occurred, further work was required to trace the root causes of the problem. In the meantime it would have been difficult for clinicians and administrators to be sure whether the problem arose from their use of the software or from a design flaw in the development of the system.

> IN SOFTWARE VF2, IF PATIENT IS SET UP IN "AUTOADMIT" MODE, PARAMETER DATA IS AUTOMATICALLY STORED IN SYSTEMS FULL DISCLOSURE DATABASE, IF PATIENT IS LATER REMOVED (BUT NOT DISCHARGED) FROM ORIGINAL ADMISSION BED/NETWORK LOCATION, DATA COLLECTION IS TEMPORARILY DEACTIVATED (FOR EXAMPLE DURING RELOCATION OR TRANSPORT TO LAB). PATIENT MAY IN FACT BE DISCHARGED AFTER DISCONNECTING MONITOR FROM NETWORK. IT IS AT THIS POINT; PATIENT DATA IS AUTOMATICALLY MOVED FROM FULL DISCLOSURE TO COMPANY'S DATABASE FEATURE (AS THIS WOULD ALSO OCCUR WHEN A PATIENT IS DISCHARGED). PROBLEM PRESENTS ITSELF WHEN A NEW PATIENT IS ADMITTED TO SAME BED/NETWORK LOCATION, BUT ORIGINAL PATIENT WAS NEVER DISCHARGED WHILE CONNECTED TO THAT LOCATION. NEW PATIENT ADMISSION BEGINS STORING DATA IN FULL DISCLOSURE DATABASE APPROPRIATELY. HOWEVER, IN PARALLEL, DATABASE INCORRECTLY BEGINS APPENDING NEW PATIENT DATA ON TOP OF OLD PATIENT'S DATA RECORD…   (MDR TEXT KEY: 1340560)

The previous incidents reveal that many usability problems stem from suppliers' and manufacturers' inability to anticipate clinical requirements. For example, the developers of the mobile patient monitoring system failed to predict that nurses would need to check how many times a battery pack had been used before making a patient transfer. The previous incident illustrates how further problems can be introduced with suppliers attempt to satisfy unanticipated clinical needs. The bug in the patient database was introduced when the initial system was rewritten to meet customer demands to record patient data as they moved from place to place in a hospital.

Communication Breakdown and Software Usability

As mentioned in the introduction, user interface design techniques often emphasize the need for designers to consult with users at a relatively early stage in the development process. They should then go back and validate any designs with those users as they move towards implementation. User testing provides feedback on whether people can actually operate the intended functionality of a system. It might be argued that manufacturers and suppliers of healthcare software have ignored these development principles given the usability problems that affect many of their products. However, things are seldom this simple. For example, the following report describes how the drug calculator of a medication assistant in a patient monitoring application would occasionally round up values to a second decimal place. The users complained that this could easily result in a medication error and that the manufacturer was failing to acknowledge the problem. The manufacturer initially responded that vigilant nursing staff ought to notice any potential problems when calculating the medication. The clinicians countered this by arguing that they had explicitly taught nursing staff to trust the calculation function as a means of *reducing human error* (see MAUDE TEXT KEY 1526689). However, the clinicians' perspective on 'usability problems' cannot always be taken at face value. Subsequent reports from the device manufacturer stressed that clinicians can configure the resolution of medication measurements through a unit manager menu:

> THIS IS BEST METHOD FOR CLINICAL STAFF, IT PRE-CONFIGURES DRUG CALCULATIONS AND ALLOWS SETTINGS TO REFLECT HOW DRUGS ARE PREPARED BY THE PHARMACY. CUSTOMER WAS TOLD, DRUG CONCENTRATION ROUNDING TO NEAREST HUNDREDTHS, COULD BE EASILY ADDRESSED IN UNIT MANAGER SETUP, TO REFLECT HIGHER RESOLUTION. THEREBY, ADDRESSING ANY CONCERN OF A ROUNDING ISSUE. manufacturer HAS REVIEWED CUSTOMER'S CONCERN AND HAVE DETERMINED THAT "DRUG CALCULATIONS" FEATURE IS FUNCTIONING AS DESIGN. ADDITIONALLY, manufacturer HAS REVIEWED WITH CUSTOMER, THE USER'S ABILITY TO CHANGE UNITS OF MEASURE, TO ACHIEVE DESIRED RESOLUTION. THE DEVICE IS PERFORMING AS DESIGNED (**MDR TEXT KEY: 1601404**)

Clinicians often complain that software does not support their basic requirements. However, many healthcare systems can be used to achieve the functionality they desire. Clinicians and technicians, typically, lack the time and opportunity to learn how to access these features. On the one hand it can be argued that clinicians should take more time to read the supporting documentation that accompanies any new device. Equally, it might be argued that the user interface design has failed if clinicians cannot work out how to access core functionality. Clinicians should be prompted to review the documentation supporting the equipment that they use and the manufacturer should assess the usability both of the device and any associated training material. Experience in user interface design has shown that such problems occur even when user testing has been performed early in the development process. For example, most mass-market operating systems have configuration options that cannot be used by the majority of consumers. More significantly, the incident illustrates the way in which usability-related adverse healthcare events often stem from a breakdown in communication between the manufacturer and the clinician. The concern is not so much that the usability problem existed in the first place; the clinicians did not know how to configure the resolution used to measure individual medications. Rather the concern is that neither of the parties involved in this incident seemed to view their dialogue as an opportunity to improve the existing design of the software application. Their focus was more on criticism and defence rather than redesign. The remedial action of retraining the end-users would not prevent other clinicians from suffering similar problems in the future.

Monitoring Bias and the Indirect Impact of Usability on Safety

The previous incident has shown that communications breakdown between manufacturer and user can lead to short-term solutions for underlying usability problems. Fortunately, national regulators and patient safety agencies are beginning to establish mishap-reporting systems that can be used both to monitor and respond to such problems. It may not be possible for these agencies to intervene directly and recommend the redesign of particular proprietary systems. It is however possible to encourage a greater emphasis on interface design during product development. For instance, the UK National Patient Safety Agency has begun to deploy a system for eliciting confidential reports of adverse medical events, including human 'errors'. It is anticipated that many of these apparent 'errors' can be traced back to usability problems with healthcare devices, including software-controlled systems. There is a paradox, however. In order to gain a better insight into the usability problems that affect clinicians and healthcare technicians, it is vital that national patient safety agencies first develop usable reporting systems. Usability problems have affected previous systems for collecting information about adverse healthcare events. For example, the FDA describes how a risk manager, JC, attempted to use their coding manual to submit a report of an incident in which a violent patient in a wheelchair was suffocated through the use of a vest restraint that was too small. The resulting classification of 1702 (amputation) and 1908 (hypertension) provided few insights into the nature of the incident:

> "She scans the list of event terms, which was detached from the rest of the coding manual... She muses: `Mr. Dunbar had OBS which isn't listed in these codes; he had an amputation which is listed; he had diabetes which isn't listed; and he had hypertension which is listed'. JC promptly enters 1702 (amputation) and 1908 (hypertension) in the patient codes. She then scans the list for Device-Related Terms... She reviews the terms, decides there was nothing wrong with the wheelchair or the vest restraint, and leaves the device code area blank." [12]

The problems of entering and coding information about adverse incidents are exacerbated when these systems are transferred from paper-based forms to computerized applications. It can be more difficult to read and navigate on-line documents than their more conventional counterparts. Careful thought must be given to the use of appropriate layouts, fonts, colours etc. Designers must also consider issues such as the display resolution of the devices that are available to those who use the forms. For example, in 2002 approximately 45% of all US Internet users had access to systems with 1024x768 resolution, 50% had 800x600 displays, 2% had 640x480. There are no comparable figures for the proportion of devices in each category within the US or UK health service. However, at least one national healthcare reporting system can only be displayed on monitors of 1024x768 resolution or higher. The same system also uses combinations of reds and greens to display the contributor's assessment of the criticality of the incident that they were reporting. Introductory courses on user interface design would advise developers to avoid such colour combinations because colour-blind users cannot easily distinguish them. The key point here is not to make arbitrary criticisms of the computer-based reporting systems that are being developed by patient safety agencies in the UK and the US. The key point is that these systems are no different from any of the other health relate software that we have described in this paper. Unless adequate resources are devoted to user interface development then clinicians and technicians will quickly abandon these applications just as they have abandoned many previous IT systems.

Conclusion and Further Work

Clinicians and support staff are faced with increasingly complex computer applications. This complexity stems, in part, from the integration of heterogeneous systems ranging from computerized patient records through to theatre management and dosage planning applications. Complexity also stems from the increased functionality offered by this new generation of IT systems. Many members of clinical staff are bewildered by the vast array of configuration options and operating modes that are supported by computer-based systems. Conversely, manufacturers often feel compelled to offer more and more software features in order to retain market position. These problems are exacerbated by the market structure for many healthcare devices in both the US and the UK. End-user requests for information about potential problems must first be relayed via suppliers and distributors before they reach the original manufacturers. This process can introduce delays and inaccuracies that further frustrate the groups and individuals who must operate healthcare-related software systems.

Usability problems are not simply inconvenient or irritating to end-users, they can also have a direct impact on patient outcomes. For example, poorly designed displays can make it difficult for clinicians to correctly read the units of a recommended dosage for particular medications. Similarly, there have been several adverse outcomes associated with systems that appended one set of records to another patient's notes [1]. Usability problems also have a number of indirect effects. For example, the costs of replacing and upgrading poorly designed software carry significant opportunity costs in terms of the systems and treatments that might otherwise have been funded. We have cited examples such as the UK Clinical Coding Information System and the US Government Computer Patient Record system both of which have absorbed significant sums of time, money and staff expertise and have yet to yield 'usable' systems.

Later sections of this paper have identified a paradox that affects the reporting of usability problems in US and UK healthcare. If clinicians and technicians cannot use the new reporting systems that are being encouraged by national patient safety agencies then there is a danger that they will be abandoned like many previous IT systems. In other words, in order to identify where usability problems exist we must first design a usable reporting system. The initial signs have not been universally encouraging. A number of prototype systems have lacked input from experienced user interface designers and have lacked end-user involvement at key stages in their development. There is, however, evidence that the US and UK patient safety agencies are aware of these problems. For example, the first human computer interaction specialist has recently been appointed to the staff of the UK National Patient Safety Agency.

The underlying message in this paper is that healthcare professionals should take a more active interest in usability engineering. This does not imply that clinicians need to become skilled in user interface design. It does, however, imply that there should be a more basic understanding of those features that are likely to make one system more usable than another. For example, the fact that one device has more operating modes that another does not necessarily mean that it will be easier to use. There is little point in paying for additional features if no one can ever learn how to use them. Unless clinicians and healthcare technicians seek more help or become better informed about human factors then there is little prospect that these insights will direct future acquisitions. For manufacturers and suppliers, there is a need to exploit user centered design techniques throughout the development cycle. Rapid prototyping and iterative testing with potential end users should guide all major design decisions. Greater care should be given to the documentation and training that are provided with increasingly complex systems. There should be a new realism about the resources of time and patience required to master software related systems. Above all, it is important to ensure that the dialogue between clinicians, technician, suppliers and manufacturers can be used to inform the subsequent development of healthcare software. All too often reports of usability problems are either dismissed as training related issues or are ignored because the device functioned as intended even if the user thought it had failed.

References

[1] C.W. Johnson, A Handbook of Incident and Accident Reporting, Springer Verlag, UK (in press, 2004).

[2] South-West Thames Regional Health Authority. Report of the Inquiry Into The London Ambulance Service Computer-Assisted Despatch System (February 1993) Original ISBN No: 0 905133 70 6

[3] K. Lyytinen and D. Robey, Learning Failure in Information Systems Development, Information Systems Journal, 9, 85-101, 1999.
http://www.ida.his.se/ida/kurser/informationssystem_forskning_utveckling/kursmaterial/learning.pdf

[4] L. Willcocks and C. Griffiths, Management and Risk in Major Information Technology Projects. In Willcocks et al (eds). Managing IT as a Strategic Resource. McGraw-Hill, London, 1997, pp.203-237.

[5] L. Willcocks, Of Capital Importance: Evaluation and Management of Information Systems Investments. Chapman and Hall, London, 1993.

[6] M.S. Bogner, Human Error in Medicine, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

[7] US General Accounting Office, Report to Congressional Committees, COMPUTER-BASED PATIENT RECORDS: Better Planning and Oversight by VA, DOD, and IHS Would Enhance Health Data Sharing, April 2001, Report number GAO-01-459, http://www.gao.gov/new.items/d01459.pdf

[8] US Food and Drugs Administration, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, U.S. Department Of Health and Human Services, Food and Drug Administration, Center for Devices and Radiological Health, Center for Biologics Evaluation and Research, January 2002. http://www.fda.gov/cdrh/comp/guidance/938.html

[9] US Food and Drugs Administration: Dick Sawyer, Do It By Design: An Introduction to Human Factors in Medical Devices, Office of Health and Industry Programs, 1996. http://www.fda.gov/cdrh/humfac/doit.html

[10] Clare-Marie Karat, Business Case Approach to Usability Cost Justification.   In R.G. Bias and D.J. Mayhew (eds.) Cost-Justifying Usability. Acaedemic Press, Boston, 1994.

[11] R. Randell and C.W. Johnson (2002) 'User adaptation of medical devices'. Proceedings of the 21st International Conference on Human Decision-Making and Control. Technical report, available on: http://www.dcs.gla.ac.uk/~johnson/eam2002

[12] US Food and Drugs Administration: M. Weick-Brady. Those codes! User Facility Reporting Bulletins, 1996. http://www.fda.gov/cdrh/issue18.pdf.