RISKS

What Happens When a Medical Office Information System Fails

Andrew Blyth
Department of Computer Studies, University of Glamorgan, Pontypridd, Mid Glamorgan, CF37 1DL, UK.
ajcblyth@glamorgan.ac.uk

As more health-care providers adopt and utilise information technology in the treatment of their patients, so the number of system failures increases. Analysis of these failures shows us that these systems are failing for a variety of technical and social reason. The question that we are forced to ask is: "Why?" In this paper I seek to highlight some of the problems that are currently facing developers and users of medical computing systems. (Editor's note: this article provides additional concrete examples of the "unintentional power" discussed in Chuck Huff's article on page 6 of this issue.)

Medical software incidents

Medical systems recalled by the FDA include the following [5]:

- A multiple-patient monitoring system was recalled because the software got patients' names mixed up with the wrong data.
- An algorithm was incorrectly programmed in a diagnostic lab instrument which caused certain patient data to be reported erroneously as all zeros.

Recalling a device is very expensive and can have dire consequences for the provider of the device. These stories of devices failing demonstrate the importance of the social context within which medical devices are used.

A digital matter of life and death

The best documented incident where medical software failed involved the Therac-25 which is a computer based electron-accelerator radiation therapy system. The Therac-25 was involved in six known incidents, including three deaths directly attributable to radiation overdoses. The incidents took place at Marietta, Georgia, USA; Hamilton, Ontario, Canada; Yakmia, Washington, USA; and Tyler, Texas, USA during a period from June, 1985 to January 1987.

Analysis of the Therac-25 identified three flaws in the system. The first was the ability of the operator to edit the command line to change the state of the machine such that the machine began to deliver radiation before the changes had taken place. The second flaw involved the safety checks being bypassed when a program counter reached zero. The third and final flaw was that certain hardware safety interlocks in the Therac-25 had been removed because those interlocks were supposed to be performed in the software. (For a detailed report on the Therac-25 see [12].)

This failure demonstrates more than any other the fact that systems failure can be attributable to failures in the development cycle of the system. Failure to correctly capture requirements, or to validate that the requirements are met in the final system, can and does lead to system failure. The developers of the system did not correctly capture and validate the method by which the system would be used. In failing to capture and validate this they failed to support or understand the users of the system.

Programming error affects hospital admissions

About 100 hospitals around the USA were forced to switch from computers to pen and paper for major bookkeeping functions because a software program could not figure out what day it was. Officials said there was no permanent loss of data or threat to treatment of patients. But the incident, apparently caused by a mistake in programming, demonstrates how institutions are accepting the risk that major disruptions might occur in the workplace as more and more functions are handed to computers. The Washington Post [13] reported:

"Problems began to appear at numerous hospitals early yesterday morning. As call after call for help arrived at SMS headquarters, technicians there realized a pattern was emerging and advised clients to shut down parts of their computer systems as they searched for the cause.

"The problem was traced some hours later to a program that allows hospitals to automate the ordering and reporting of laboratory tests. Due to a fault in the aging software, the machines were unable to accept as valid the date September 19, 1989, and went 'into a loop,' refusing to work, spokesman A. Scott Holmes said.

"By day's end, computer services at about 100 of SMS's 600-700 client hospitals had been disrupted."

Software failure may be behind ambulance crisis

The references [9, 10] report on the Software failure of the Computer Aided Dispatch System. In this article computer specialists say that the system blamed for the crisis at the London Ambulance Service appeared to ignore basic tenets for software where breakdown would put lives at risk. The failure of the computer system for over 36 hours on a Monday and Tuesday, which was said to have cost between 10 and 20 lives, raised serious questions about the way it was designed and tested. The investigators concluded:

• that insufficient time and resources where spent on getting the users of the system involved in its development, and training them on how to use the new system.

• that at the time that the system went live, the software was incomplete and had not been properly tested.

• that the decision not to have a back up of any kind of the system was unwise. The result of not having a back up was that whenever the system fell over, all information on logged calls and ambulance crew allocation was lost.

The report also noted that the ambulance and central control staff had no confidence in the system and were not all fully trained [10]. Analysis of this case study demonstrates the need to get users and stakeholders involved in system development and to train and support the user of the system. In addition, this case study demonstrates that information technology with complex organisations can only work when it is woven and supported within the social fabric of the organisation.

No laughing matter

In [8] it is reported that a 13 year-old daughter of a hospital records clerk in Jacksonville, Fla, USA., used her mother's computer during an office visit to print out names and numbers of patients previously treated in the hospital's emergency room. According to police, the girl then telephoned seven people and falsely told them they were infected with the HIV virus. One person attempted suicide after the call. Upon arrest the girl told the police that the phone calls where just a prank.

This story demonstrates that concepts like security and privacy are social in nature, and that to consider them in purely technical terms is to invite disaster.

Hacker nurse makes unauthorised changes to prescriptions

In [7] it was reported that a male nurse was convicted of hacking into a hospital's computer system and modified entries, including prescriptions. The hacker:

• prescribed drugs normally used to treat heart disease and high blood pressure to a 9 year old with meningitis. This change was spotted by a ward sister;

• prescribed antibiotics to a patient in a geriatric ward. These drugs were administered to the patient, with no apparent adverse reaction;

• "scheduled" an unnecessary X-ray for a patient;

• "recommended" a discharge for another patient.

The hacker gained access to the computer system after learning the password through observing a doctor who was having trouble logging in. He qualified as a nurse in 1989. He is reported to have undergone a considerable personality change as the result of a road accident in 1984. As well as developing a fascination for computers and other hi-tech equipment, he had apparently developed a "lack of sensitivity to the consequences of his actions".

He had been sacked for unprofessional behaviour in 1990, but was re-employed in 1992 at the same hospital.He pleaded guilty to unauthorised modification of computer records. He offered no explanation for his actions, but denied any malicious intent. He was jailed for 12 months.

This small case study in the applicability of the Computer Misuse Act of 1990, UK, demonstrates that in the medical sector security and privacy are only as good as the people who use the system. In short, the only way to guarantee security and privacy of information contained on a system is to let no one use the system.

Conclusions

In [1,2,3] there are detailed examples of what happens when computer systems and their development processes fail. In this paper I have not sought to provide solutions, but rather to highlight some of the problems facing medical computng. In addition, within this paper I have also sought to shown through the case-studies that information technology within the health-care arena can not be considered outside of its social and organisational context, and that to do so exposes people to possible lethal risks.

There is growing evidence to suggest that systems are failing for social and organisational reasons [4, 6, 10]. In [11] the point is made that we need to integrate systems into the social fabric of organisations if the system is not to fail. I believe that all of the above examples of system failure demonstrate that we need to view and understand the human and social components of a medical computing system if that system is to stand a chance of not failing.

References

[1] Lauren Wiener, Digital Woes, Addison Wesley, 1993.

[2] Peter Neumann, Computer Related Risks, Addison Wesley, 1995.

[3] Nancy Leveson, Safeware: System Safety and Computers, Addison Wesley, 1994.

[4] J. Glasser, "Organisational Aspects of System Failure: A Case Study at the Los Angeles Police Department", in Proc of the 2nd International Conference on Information Systems, 1981.

[5] H. Bassen, et al., "Computerized Medical Devices: Usage Trends, Problems, and Safety Technology," in Proc. 7th Annual Conference of IEEE Engineering in Medicine and Biology Society, 1985.

[6] S. B. Sloane, "The Use of Artificial Intelligence by the United States Navy: Case Study of a Failure", AI Magazine, Spring, 1991.

[7] The Guardian Newspaper. UK, 21st December 1993.

[8] Robert Fox, "NewsTrack", in Communications of the ACM, 38(5), May 1995.

[9] The Independent Newspaper. UK, 30 Oct, 1992.

[10] Report of the Inquiry Team into the London Ambulance Service, Communications Directorate, Feb 1993, ISBN 0-905133-70-6.

[11] F. W. Wolek, "Implementation and the Process of Adopting Managerial Technology", Interfaces, 5(3), 1974.

[12] Nancy G. Leveson and Clark S. Turner, "An Investigation of the Therac-25 Accidents", IEEE Software, 26(7), July 1993.

[13] "Sick Software Checks in at 100 Hospitals." The Washington Post, Wednesday, 20 September 1989, page F-1.