Appendix

Cost of Missing Quality: Case Studies

CASE 1: SOCIAL SECURITY TO PAY $500 MILLION TO 80,000 VICTIMS OF DATABASE ERROR

The Washington Post, one of the main newspapers in Washington D.C., cites in Reference 1 the case of a decision made based on a computer system error that caused 80,000 people to stop receiving their monthly benefits. It all started in 1996 when a federal law was promulgated to prevent justice fugitives from receiving any benefit from the government. In order to implement the measure, information from several government databases was compiled and, based on that, several thousand people had their benefits cancelled. The measure helped to capture several justice fugitives, but it also affected several people who had no criminal records. The article cites the case of Rosa Martinez, a resident from Redwood City California, whose monthly disability check was cancelled because the system had confused her records with those of another Rosa Martinez from Miami Florida. Apart from the hardships caused to people affected by the mistakes, the government faced several lawsuits that resulted in a considerable amount of vindications.

System Characteristics

The system described in the case study has typical components of a decision support system: data sources and a series of rules to analyze the data and recommend actions based on it. In this case, the quality of the data and the quality of the implemented algorithms to analyze it were not good enough to prevent problems for the system's owner.

Software Quality Engineering: A Practitioner's Approach, First Edition. Witold Suryn.

Published 2014 by John Wiley & Sons, Inc.

Missing Quality

- Functionality. The system was not able to correctly perform its task. The data used in the system did not contain enough information to identify in a conclusive way every person in the databases. Due of this reason, when information from several data sources was aggregated, the implementers very likely had to resort to manual matching or heuristic algorithms.
- Testability. Even though the newspaper article does not mention this information, the outcome suggests that the system designers did not have enough elements to be able to correctly match the data from different data sources and be able to test different scenarios. Its lack of testability is very likely the cause of several of the errors when government policies changed.

Impact

Level B: Important economic or social impact.

- Privations to a vulnerable group of people. Arguably, the biggest impact caused by the missing quality in this system was the damage caused to the people who did not receive their benefits. The group of people who were entitled to receiving government benefits is comprised of the most vulnerable individuals in society, the elderly and the disabled. The disappearance of a source of income for these persons very likely caused them considerable damage.
- Liability and exposure costs. The government exposed itself to lawsuits by the affected people. These lawsuits caused a drain of financial resources that had to be rerouted from other sources, both to cover the costs of litigation and the indemnifications.
- Correction costs. Apart from the financial costs associated with the legal actions, the government had to dedicate resources to fix and verify the decisions made by the system, to prevent further errors and fix the existing ones.
- Unhappy customers. Even though in this case the error did not come from a business, there were unhappy customers, or in this case, unhappy citizens. The error might have political impact as the knowledge about the mistake might have led to loss of confidence in the government.

CASE 2: VA WRONGLY TELLS VETERANS THEY HAVE A FATAL DISEASE

The Federal Computer Week Magazine [2] wrote, in an article dated August 26, 2009, of a case in the Veterans Affairs Department of the United States (an organization that administers benefits for military veterans), where an error in a decision support system caused 1,200 veterans to be wrongly informed that they had acquired a fatal disease. The following text taken from the magazine describes the problem:

> The Veterans Affairs Department sent electronically generated letters last week that wrongly told as many as 1,200 veterans they have been diagnosed with the fatal Lou Gehrig's neurological disease, according to Jim Bunker, president of the National Gulf War Resource Center, a veteran's services nonprofit group. The VA did not comment on the cause of the mistake. Bunker said it happened due to years of misapplication of computer medical coding. Bunker said that for many years, the VA applied a medical code to refer to undiagnosed neurological disorders. Several years ago, he said, VA expanded the code category to include amyotrophic lateral sclerosis, commonly called Lou Gehrig's disease. Recently, the VA determined ALS to be a service-connected disability and generated automatic letters to all veterans whose records included the code for the disease. However, since the coding contained both ALS and undiagnosed neurological disorders, some of those letters were erroneous, Bunker said.

System Characteristics

The system described in the article is a medical decision support system. The error itself was not caused by a problem in the computer code or the data analysis algorithms, but by a human mistake in the way the data was being categorized. As mentioned in the case description, the problem was that two different diseases were being saved in the databases as if they were the same.

Missing Quality

- User error protection. The root cause was that the system allowed an incorrect categorization of the data by the people responsible for entering the data.
- Suitability. The category used to "tag" the new disease was not "suitable" to tag the new disease. In this case the lack of suitability was not a defect in the software, but in the data. Since all decision support systems rely heavily on databases, they are particularly vulnerable to database errors.

Impact

Level C: Significant impact on human life.

- The system gave incorrect information to clients in sensitive matters. The impact of the lack of reliability in this case did not financially affect the organization that caused the error, but it affected the well being of the recipients of the letters, since they received wrong information about their health.

CASE 3: SEVERAL PERSONS DIE DUE TO A BUG IN SOFTWARE IN THE THERAC-25 MACHINE

The Therac-25 machine is a case of an industrial support system that had lethal consequences due to errors in software. The Therac-25 was a radiation machine used in hospitals to perform radiation therapy and create X-ray pictures of patients. It was involved in a series of accidents that caused deaths and severe injuries to people who were exposed to the machine. The following description of the problem was taken from Reference 3:

The machine offered two modes of radiation therapy:

- direct electron-beam therapy, which delivered low doses of high-energy (5 MeV to 25 MeV) electrons over short periods of time
- megavolt X-ray therapy, which delivered X-rays produced by colliding high-energy (25 MeV) electrons into a "target"

When operating in direct electron-beam therapy mode, a low-powered electron beam was emitted directly from the machine then spread to safe concentration using scanning magnets. When operating in megavolt X-ray mode, the machine was designed to rotate four components into the path of the electron beam: a target, which converted the electron beam into X-rays; a flattening filter, which spread the beam out over a larger area; a set of movable blocks (also called a collimator), which shaped the X-ray beam; and an X-ray ion chamber, which measured the strength of the beam.

The accidents occurred when the high-power electron beam was activated instead of the intended low power beam, and without the beam spreader plate rotated into place. The machine's software did not detect that this had occurred, and therefore did not prevent the patient from receiving a potentially lethal dose of radiation. The high-powered electron beam struck the patients with approximately 100 times the intended dose of radiation, causing a feeling described by patient Ray Cox as "an intense electric shock." It caused him to scream and run out of the treatment room. Several days later, radiation burns appeared and the patients showed the symptoms of radiation poisoning. In three cases, the injured patients died later from radiation poisoning.

System Characteristics

The Therac-25 machine is a specific example of a system in which one of the parts is an industrial support system. In this case the software part was responsible for controlling the hardware parts of the machine and the malfunction of the software had lethal consequences.

Missing Quality

- Accuracy. The main cause of the accidents was the lack of accuracy in the doses of radiation that the patients had to receive. As mentioned in the problem description, the patients received up to ten times the amount of radiation that they should have received.

- Testability. According to a study made about the case cited in Reference 4, a commission formed to investigate the case found that one of the main causes of the problems was that the software was designed in such a way that it was impossible to test. It is also mentioned that the machine was never tested with the exact configuration of hardware and software where it was going to run until it was already in the hospital
- Fault tolerance. The machine failed only when a nonstandard sequence of keys was pressed. This incorrect sequence of keys can be considered as a "fault" from the part of the operator, so the software was not prepared to handle or "tolerate" this kind of errors.

Impact

Level A: Death of human being.

- Physical damage to patients. The machine caused radiation poisoning and burns to several patients exposed to it.
- Death of patients. The deaths of six people were proved to be directly caused by the machine.
- Investigation costs. Several experts of different fields had to be engaged in the investigations that led to the determination of the causes of the problems with the machine.
- Recall costs. After it was confirmed that machine could harm the people it was supposed to help, all the Therac-25 machines were recalled by the manufacturer.

CASE 4: SMELT PLANT SHUTS DOWN DUE TO SOFTWARE BUG

The Website of the computer science department of the Tel Aviv University [4] cites a note in a New Zealand newspaper that describes how a smelting plant was shut down due to errors in software. The exact transcription of the problem is the following:

> A computer software error at the Tiwai Point aluminum smelter in Southland, New Zealand at midnight on New Year's Eve 1997 caused more than $AU 1 million of damage. The software error was the failure to account for leap years (and considering a 366th day in the year to be invalid), causing 660 process control computers to shut down and the smelting pots to cool. The same problem occurred two hours later at Comalco's Bell Bay smelter in Tasmania (which is two hours behind New Zealand). The general manager of operations for New Zealand Aluminum Smelters, David Brewer, said "It

> was a complicated problem and it took quite some time [until midafternoon] to find the cause. (Originally from *The New Zealand Herald*, January 8, 1997, and *The Dominion*, in Wellington, New Zealand).

System Characteristics

The system in this case study was responsible for controlling the operations of a smelting plant. It can be classified in the control systems branch of the industrial support systems. According to the information presented in the note, the main cause of the problem was incorrect date handling in the internal system's calendar.

Missing Quality

- Accuracy. In this case study, the root cause of the problem was the lack of accuracy in date handling in the software that controlled the smelting plant. There are not many details available about the causes of the error, but it is very likely that the software programmers had implemented their own date format, instead of using an existing and tested one.
- Fault tolerance. The whole system was affected by a problem in the handling of the dates. While it is very hard to prevent unknown errors, there are development techniques, such as defensive programming, that can help to minimize the effect of faults. In this case, it is unlikely that such a technique was used.

Impact

Level B: Dramatic economic impact on the company.

- Financial loss. The company lost $AU 1 million due to the problem.

CASE 5: VOYAGES-SNCF.COM

Voyages-sncf.com is a popular e-business site in France. It's used by 25% of the French population. On November 20, 2008, this site was inaccessible for 30 hours. The problem occurred after a new version of the site was deployed [5].

Missing Quality

- Fault tolerance. It is probable that the new version of the system was not able to handle errors; this problem caused the site to crash due to events that normally would not bring down the complete site. As a result of this, many customers were affected and the site lost revenue.

- Installability. Again, it is probable that installing a new version did not go as required, creating a nonfunctional instance of the system.
- Coexistence. One of the frequent reasons for "explosions" of new releases of the software is forgetting that the software usually resides on an operating system that contains other applications. The modifications in new releases sometimes put in conflict functionalities that worked well before the modifications.

Impact

Level C: Critical economic impact.

- 30 hours of downtime, considerable revenue lost, dissatisfied customers.

CASE 6: EBAY

On June 14, 1999 the eBay web site was down for 22 hours, and 2.3 million bids failed. The web site claims being a victim of first "hardware failure" and after that "network failure" [6].

Missing Quality

- Fault tolerance. The system was not able to properly react to failures and continue working.

Impact

Level C: Critical economic impact.

- 2.3 million bids failed.

CASE 7: A SOFTWARE GLITCH ERASED THE WAREHOUSE'S EXISTENCE TO THE BRITISH FOOD RETAILER SAINSBURY'S

In October 2004, the automated distribution system of the British food retailer Sainsbury's had a problem that affected the distribution of merchandise between stores. The problem resulted in the system sending the merchandise to wrong locations. The company had to disregard the system and lost the investment of $526 million it had made in it [7].

Missing Quality

- Accuracy. The lack of accuracy in the data coming out of the system caused the deliveries to be sent to incorrect locations.

Impact

Level B: Major financial impact.

- $526 million lost.

CASE 8: HIT THE WRONG KEY, BECOME A VERB . . .

Pablo Davila lost at least $207 million of Codelco, a state-owned Chilean company by typing the wrong financial transaction into his computer. He typed "buy" when he says he meant to type "sell." Now, all of Chile is obsessed with the mistake that cost 0.5% of Chile's GNP and the new word "davilar" is a verb that is ". . . loosely translated as 'to botch things up miserably.'" Integral text from Reference 8.

Missing Quality

- Operability, user error protection. The system does not support the user to operate and control the software, and if he or she makes an accidental manipulation, the system does not prevent it from happening.

Impact

Level B: Human lives and financial impact.

- Lost at least $207 million.

CASE 9: ONLINE BANKING SYSTEM FAILURE

Three of the twelve largest banks in Japan merged. The results were not pretty, including "more than 30,000 transaction errors and 2.5 million delayed debits" and "2.5 million of the 3 million automatic debits scheduled to be processed on 1 Apr. 2002, including utility and credit card bills, couldn't be made on that day."

The problem was that each of the banks ran a different system (Hitachi, IBM, and Fujitsu, although no software was mentioned). They built some integration glue, but it did not work. About 30,000 incorrect double withdrawals and about 5,000 double deposits were found and corrected.

According to the bank, "the overall accumulation of delayed transaction would need the whole week to finish." Integral text from [9].

Missing Quality

- Interoperability.

Impact

Level B: Critical economic impact on Japan's banking system.

- About 30,000 incorrect double withdrawals
- About 5,000 double deposits
- Accumulation of delayed transaction.

CASE 10: MAJOR COMPUTER FAILURE AT HSBC BANK

From August 15 to August 22, 2011, 4 million U.S. customers of HSBC Bank could not use their debit cards or their deposits were delayed [10].

Missing Quality

- Fault tolerance
- Maturity.

Impact

Level D: Negligible economic or social impact.

- Customer satisfaction lost.

CASE 11: BOEING 787 NETWORKING ISSUES

Boeing's new 787 Dreamliner passenger jet may have serious security vulnerability in its onboard computer networks that could allow passengers to access the plane's control systems, according to the U.S. Federal Aviation Administration. The computer network in the Dreamliner's passenger compartment, designed to give passengers in-flight Internet access, is connected to the plane's control, navigation, and communication systems, an FAA report reveals. The revelation is causing concern in security circles because the physical connection of the networks makes the plane's control systems vulnerable to hackers. A more secure design would physically separate the two computer networks. Boeing has said that it is aware of the issue and has designed a solution it will test shortly [11].

System Characteristic

The described system is an airplane network system. It performs the functions of giving passengers in-flight Internet access, but is connected to the plane's control, navigation, and communication systems.

Missing Quality

- Security. No isolation between the plane control network and the entertainment network. This situation could lead to possible failures or faults in the network system and allow access to the plane's control systems by passengers.

Impact

Level A: Probable loss of human lives.

- Possible airplane operation failures
- Flight pilot may lose airplane control
- Could lead to loss of human lives.

CASE 12: BUFFER OVERFLOW IN BERKELEY UNIX FINGER DAEMON

The first Internet worm (the so-called Morris Worm) infected between 2,000 and 6,000 computers in less than a day by taking advantage of a buffer overflow. The specific code is a function in the standard input/output library routine called gets(), designed to get a line of text over the network. Unfortunately, gets() has no provision to limit its input, and an overly large input allows the worm to take over any machine to which it can connect.

Programmers respond by attempting to stamp out the gets() function in working code, but they refuse to remove it from the C programming language's standard input/output library, where it remains to this day [12].

System Characteristic

Personal or business computers connected to the Internet.

Missing Quality

- Security. Serious vulnerability in the software library that allows malicious software to take control computers.

Impact

Level C: Some significant economic impact.

- Infection by a computer worm
- Possible data theft
- Possibility of malicious deeds.

CASE 13: AT&T NETWORK OUTAGE

A bug in a new release of the software that controls AT&T's #4ESS long distance switches caused these mammoth computers to crash when they received a specific message from one of their neighboring machines—a message that the neighbors send out when they recover from a crash.

One day a switch in New York crashed and rebooted, causing its neighboring switches to crash, then their neighbors' neighbors, and so on. Soon, 114 switches were crashing and rebooting every six seconds, leaving an estimated 60 thousand people without long distance service for nine hours. The fix: engineers load the previous software release [13].

System Characteristic

Telecommunication long distance switches.

Missing Quality

- Recoverability. Inability to recover automatically from a crash.
- Maturity. Inability to stay in functional mode after receiving a special message.

Impact

Level B: Continuous usage of the system.

- Long-distance service disruption.

CASE 14: THE PING OF DEATH

A lack of sanity checks and error handling in the IP fragmentation reassembly code makes it possible to crash a wide variety of operating systems by sending a malformed “ping” packet from anywhere on the Internet. Most obviously affected are computers running Windows, which lock up and display the so-called “blue screen of death” when they receive these packets. But the attack also affects many Macintosh and Unix systems as well [14].

System Characteristic

Computer base communication system.

Missing Quality

- Fault tolerance. Operating systems unable to continue to operate after receiving malformed messages.

Impact

Level C: Continuous usage and possible significant economic loss.

- Vulnerability to malicious attack.

CASE 15: TELECOMMUNICATIONS FAILURE ISOLATES LIBERIA

“Monrovia—Liberia, entered on Wednesday the seventh day of a major breakdown in its telecommunications facilities, isolating the country from the international community and impeding local transactions.” Telephone, fax, and telex services of the Liberia Telecommunication Corporation went mute, leaving internal and external communications to two private phone and Internet companies. The managing director of the corporation, Charles Roberts, told PANA that the breakdown was due to loss of essential data from the memory bank of the main central processing unit of the switching exchange. Roberts admitted the company’s equipment was near obsolete, but rejected claims that neglect of maintenance had caused a collapse in the facility [15].

Missing Quality

- Fault tolerance. Operating systems unable to stay up after data loss.
- Suitability. Ability to backup data used for system processing.

Impact

Level A: Disastrous economic or social impact.

- Major economic impact on Liberia
- Business and bank activities paralyzed
- Local and international institutions practically paralyzed.

CASE 16: FAILURE OF LONDON AMBULANCE DISPATCH SYSTEM

In 1992, the failure of the London ambulance system's information management system resulted in deaths of several people, since no ambulance arrived on time. The following description of the situation was taken from Reference 16:

> After a whole slew of issues, including a project cancellation and re-design, a software system got developed and was deployed the morning of October 26, 1992. Just a few hours later, however, problems began to arise. The AVLS was unable to keep track of the ambulances and their statuses in the system. It began sending multiple units to some locations and no units to other locations. The efficiency with which it assigned vehicles to call locations was substandard. The system began to generate such a great quantity of exception messages on the dispatchers' terminals that calls got lost. The problem was compounded when people called back additional times because the ambulances they were expecting did not arrive. As more and more incidents were entered into the system, it became increasingly clogged. The next day, the LAS switched back to a part-manual system, and shut down the computer system completely when it quit working altogether eight days later.

System Characteristics

The London dispatch system can be classified as a hybrid between an information management system and a telecommunications control system, because it manages ambulance information and it has remote communication capabilities to allow communications between ambulances and central.

Missing Quality

- Accuracy. The LAS was not able to adequately track and manage of the ambulances and the service calls locations.

Impact

Level B: Major financial impact.

- Loss of human lives. According to Reference 16, at least 46 people died who could have been saved.

- Financial loss. The system was shut down completely, which caused complete investment loss.

CASE 17: TORONTO PUBLIC HEALTH COMPUTER ACCIDENTALLY ERASES RECORDS

The Toronto Star reported in its March 10, 2003, issue [17] a problem in an information management system that caused the loss of medical information. The problem description is the following:

> A computer fault may have accidentally erased the immunization records of thousands of Toronto school children, the city's public health department fears. Last April, the department discovered that its immunization records information system was erasing files from among 425,000 student records, Dr. Barbara Yaffe, associate medical officer of health, said. "It appears it was randomly erasing files—and we don't know how many."

The department tried to obtain technical help from the provincial health ministry, but its technicians were among the 45,000 Ontario civil servants taking part in a 54-day strike that spring.

System Characteristic

The described system is a medical information management system. Such systems are common in the hospitals and clinics and are used to make decisions regarding a patient's health.

Missing Quality

- Reliability. The system could not guarantee the integrity of the stored information.
- Functionality. Due to its lack of integrity, the system was not able to accomplish is main task.

Impact

Level C: Minor impact on human lives.

- Rework costs. The missing data had to be reentered manually in the system; in addition, all the information had to be verified to ensure record accuracy.
- Possible impact to the health of the affected children. Due to missing information, some children could have been vaccinated more than one time, which could have had negative effects on their health.

REFERENCES

1. Gowen A. “Social Security to Pay $500 Million to 80,000 Victims of Database.” *The Washington Post*, August 12, 2009.
2. Lipowicz A. “VA Wrongly Tells Vets They Have a Fatal Disease.” *Federal Computer Week*, August 26, 2009. Available at http://fcw.com/articles/2009/08/26/va-erroneously-informs-vets-of-fatal-disease-diagnosis.aspx. Accessed May 10, 2013.
3. “An Investigation of the Therac-25 Accidents.” *IEEE Computer* 1993; 26(7):18–41.
4. “Software Horror Stories.” Entry 55. Available at http://www.cs.tau.ac.il/∼nachumd/horror.html. Accessed May 10, 2013.
5. “Retour à la normale pour Voyages-sncf.com.” *01net*, November 20, 2008.
6. Clark T. “eBay Online Again after 14-Hour Outage.” *CNET*, August 6, 1999.
7. Leveson N. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.
8. Wayner P. “Hit the Wrong Key, Become a Verb.” *Risk Digest* 1994; 15(66).
9. Ishikawa I. “Online Banking System Failure in a Big Way.” *Risk Digest* 2002; 22(3).
10. Osborne H. “HSBC Computer Failure Leaves Customers Short of Cash.” *The Guardian*, November 4, 2011.
11. Zetter K. “FAA: Boeing’s New 787 May Be Vulnerable to Hacker Attack.” *Wired*, April 1, 2008.
12. Panettieri JC. “Who Let the Worms Out?” *eWEEK*, March 12, 2001.
13. “Software Glitch Cripples AT&T.” *Telephony*, January 22, 1990.
14. Erickson J. *Hacking: The Art of Exploitation*, 2nd ed. No Starch Press, 2008.
15. “Telecommunications Failure Isolates Liberia.” *Panafrican News Agency*, November 1, 2000.
16. Dalcher D. “Disaster in London: The LAS Case Study.” In *ECBS 99 IEEE Conference and Workshop on Engineering of Computer-Based Systems, March 7–12,* 1999, pp. 41–52.
17. Smith C. “Health Records Feared Erased.” *Toronto Star*, March 10, 2003.