‘WE DID NOTHING WRONG’

CASE

109

A DISSECTION

WHY SOFTWARE QUALITY MATTERS

As software spreads from computers to the engines of automobiles to robots in factories to X-ray machines in hospitals, defects are no longer a problem to be managed. They have to be predicted and excised. Otherwise, unanticipated uses will lead to unintended consequences. For proof, look no further than the cancer patients in Panama who died after being overdosed by a Cobalt-60 radiotherapy machine. Or ask the technicians who plugged data into the software that guided that machine, and are now charged with second-degree murder.

BY DEBORAH GAGE AND JOHN McCORMICK

Additional reporting by Berta Ramona Thayer in Panama

PHOTOGRAPH BY NITIN VADUKUL

PANAMA'S CANCER INSTITUTE BASE CASE

Organization: National Cancer Institute
Headquarters: Gorgas Hospital, Ancon Hill, Panama Canal Zone, Panama
Mission: To treat and cure patients with cancer.
Director: Juan Pablo Bares, M.D.
Financials: Funded by the Panamanian government.
Challenge: Find resources to treat a patient load that has more than quadrupled since 1997, with 10 to 15 new patients a day.
BASELINE GOALS:

- Improve quality-assurance programs, to avoid accidental patient deaths or injuries.
- Install third linear accelerator, to shorten waiting list for cancer treatments.
- Eliminate subsidies paid to private hospitals for overflow work, which has cost the government $10 million over the past three years.

Victor Garcia considers himself lucky to be alive. Three years ago, a combination of cancer and miscalculation almost killed him.

The former distribution manager for fragrance maker Chanel now can feel the hot Panamanian morning sun stream through his living-room window. He can smell lunch cooking in the kitchen. He can sit in an armchair surrounded by pictures of his six children and six grandchildren and talk to his wife. Simple pleasures he almost lost following a software malfunction. In November of 2000, Garcia and 27 other patients at the National Cancer Institute in Panama were jolted with massive overdoses of gamma rays partly due to limitations of the computer program that guided use of a radiation-therapy machine.

In the 40 months that have passed, 21 patients have died. While it's unclear how many of the patients would have died of cancer anyway, the International Atomic Energy Agency (IAEA) said in May 2001 that at least five of the deaths were probably from radiation poisoning and at least 15 more patients risked developing "serious complications" from radiation.

Garcia, being treated for prostate cancer, survived but suffered damage to his intestines. He now has a colostomy. "I am very lucky," he says, shaking his head in wonderment. "That's what the [investigating] doctors from Houston told me. 'You are so lucky.' "

The three Panamanian medical physicists who used the software to figure out just how much radiation to apply to patients are scheduled to be tried on May 18 in Panama City on charges of second-degree murder. Under Panamanian law, they may be held responsible for "introducing changes into the software" that led directly to the patients' deaths, according to Special Superior Deputy Prosecutor Cristobal Arboleda.

The physicists, of course, thought they were helping the patients. Having consulted a doctor at the hospital and the software's manual, they thought they had figured out how to place five radiation shields over each patient's body, instead of four, to protect against possible overdoses. "I thought I was home free," one of them, Olivia Saldaña, recalls now.

This is not a cautionary tale for medical technicians, even though they can find themselves fighting to stay out of jail if they misunderstand or misuse technology. This also is not a tale of how human beings can be injured or worse by poorly designed or poorly explained software, although there are plenty of examples to make the point. This is a warning for any creator of computer programs: that software quality matters, that applications must be foolproof, and that—whether embedded in the engine of a car, a robotic arm in a factory or a healing device in a hospital—poorly deployed code can kill.

In this case, a St. Louis company, Multidata Systems International, has found itself in and out of courts in two countries for much of the past three years, fending off charges that its product is at fault in a score of fatalities. The deaths occurred more than 2,000 miles from its home, at an installation of a customer it claims it did not even know it still had—until the death toll began mounting.

Now Multidata may face judgments that could damage—if not destroy—the company itself, if the firm is found guilty and is forced to pay damages sought by the victims. No one can accurately predict the amount Multidata would have to pay if the victims succeed in suing in the U.S. So far the plaintiffs have failed. But each of the 28 victims could be entitled to as much as $500,000 to $1 million of compensation for such factors as pain and suffering, lost wages and the number and age of surviving dependents, according to Brian Kerley, a defense attorney at a leading New York malpractice firm. Using those numbers, Multidata could be facing total damages in the range of $14 million to $28 million. Multidata, which is privately held, says it has about $2 million in annual sales and fewer than 15 employees.

That company could just as well be your company, whether you write software in small or large teams; and whether you operate domestically or in multiple nations in a rapidly globalizing economy. You are at risk if you place your product in conditions where human lives are at stake. Indeed, it's not the first time that software has been a suspect in a series of unexpected fatalities.

- In the mid-1980s, poor software design in another radiation machine, known as the Therac-25, contributed to the deaths of three cancer patients. The Therac-25 was built by Atomic Energy of Canada Ltd., which is a Crown corporation of the government of Canada. In 1988, the company incorporated and sold its radiation-systems assets under the Theratronics brand. There does not appear to be any formal investigation of the Therac-25 accidents, but according to an in-depth examination by Nancy Leveson, now a professor at the Massachusetts Institute of Technology, and the accounts of other software experts, the design flaws included the inability of the software to handle some of the data it was given; and the delivery of hard-to-decipher user messages. In a twist of fate, Theratronics, which was ultimately acquired by the Canadian life-sciences company MDS, manufactured the radiation-therapy machine used at the cancer institute in Panama.
- In February 1991, during Operation Desert Storm, an Iraqi SCUD missile hit a U.S. Army barracks in Saudi Arabia, killing 28 Americans. The approach of the SCUD should have been noticed by a Patriot missile battery. A subsequent government investigation found a flaw in the Patriot's weapons-control software, however, that prevented the system from properly tracking the missile. More recently, during Operation Iraqi Freedom, the Patriot missile system mistakenly downed a British Tornado fighter and, according to the *Los Angeles Times* and other reports, an American F/A-18c Hornet. The pilot in the single-seat Hornet and the two crew members aboard the British jet were killed. The incidents are still under investigation, but Pentagon sources familiar with the Hornet incident told the *L.A. Times* that investigators were looking at a glitch in the missile's radar system that made it incapable of properly distinguishing between a friendly plane and an enemy missile. Raytheon, the maker of the Patriot missile system, did not want to comment on the 1991 incident. It also said the government was still investigating the more recent incidents and that reports the software may be at fault were "off base."
- A software glitch was cited in a Dec. 11, 2000, crash of a U.S. Marine Corps Osprey tilt-rotor aircraft, in which all four Marines on board were killed. According to Marine Corps Maj. Gen. Martin Berndt, who presented the finding from a Judge Advocate General investigation, "the mishap resulted from a hydraulic-system failure compounded by a computer-software anomaly." A hydraulic line broke in one of the craft's two engine casings as the pods were being moved from airplane mode to helicopter mode in preparation for landing. When the flight-control computer realized the problem, it stopped the rotation of the engine pods. The pilots, trained to respond, tried to reset the pods by pressing the primary reset button, but the finding stated that a glitch caused "significant pitch and thrust changes in both prop rotors," which led to a stall. The plane crashed in a marsh. The craft is made by a partnership of Boeing and Bell Helicopter. A Boeing spokesman said changes were made in the software but referred requests for details about the software anomaly to the government.

THE **PLAYER** ROSTER

Olivia Saldaña
Physicist, National Cancer Institute (NCI), Panama
Saldaña is one of three physicists charged with second-degree murder in Panama for entering data into Multidata's software that produced inaccurate amounts of time for patients to be treated with a Cobalt-60 beam. She continues to work at the hospital because, she says, "If we did not work, the patients would die."

PANAMANIANS
Victor Garcia
Retired businessman
Garcia is one of seven cancer patients who survived the radiation overdoses at the institute in 2000. Overall, 21 patients have died. He is a party to lawsuits against Multidata International Systems and MDS, the owner of the Cobalt-60 therapy machine, in both Panama and the U.S.

Dr. Juan Pablo Bares
Director, NCI
Bares sought international help to understand the causes of the overdoses after the hospital realized in March of 2001 they had occurred. Bares offered to resign after the overdoses became public in 2001, but the hospital board refused to accept his resignation.

Camilo Jorge
Services manager, ProMed
ProMed solicited a bid from Multidata on a referral from General Electric Medical Systems because the NCI couldn't afford the treatment-planning software offered with the Cobalt-60 teletherapy machine. Jorge says it is the only time ProMed has done business with Multidata.

Cristobal Arboleda
Special Superior Deputy Prosecutor
Arboleda led the investigation into the causes of the overdoses for Panama's Ministry of Health and is now prosecuting the physicists. He says his office had little experience with software and his staff has had to learn on the job.

MULTIDATA SYSTEMS INTERNATIONAL
Mick Conley
General business manager
Conley, a 13-year veteran of the radiation-therapy systems company, oversees product sales and marketing. He's unflappable when pressed about the role of the company's software in the radiation accidents in Panama and maintains that they would not have happened if the staff at the NCI had followed the manual and verified the software's results before treating patients.

Arne Roestel
President
Roestel runs the privately held company, which he founded in 1979. He's been working in the radiation-treatment software industry since the late 1960s. Business manager Conley calls him a pioneer in the field and says he was one of the first people in the country to work on computerized radiation-treatment systems.

OFFICE OF COMPLIANCE, CENTER FOR DEVICES AND RADIOLOGICAL HEALTH (CDRH), FOOD AND DRUG ADMINISTRATION
Timothy Ulatowski
Director
A 30-year veteran of the FDA, and one of the few people to wear a tie in the CDRH office, Ulatowski is a direct, to-the-point manager who oversees a number of FDA operations, including enforcement of medical-device and radiological-health laws and regulations. He holds a B.S. in microbiology and an M.S. in biomedical engineering.

John Murray
Software and Part 11 compliance expert
Murray is the CDRH's primary advisor on all aspects of software, including validation, policy, and classification. He holds an undergraduate degree in electrical engineering and a graduate degree in computer science; he can explain the complexities of medical devices and their software in lay terms.

A spokesman for the Navy's Air Systems Command, which investigated the incident, confirmed the software problem, but was not able to provide additional details.

Nor are these incidents likely to be the last. In 2002, the Food and Drug Administration (FDA), which oversees medical-device software, said of 3,140 medical-device recalls conducted between 1992 and 1998, 242, or 7.7%, were attributed to software failures. The FDA also says the number of software-related recalls may be underreported because it's often hard to determine the exact cause of a problem in the immediate aftermath of an accident.

There's a financial cost to all organizations that use badly designed and deployed software as well. Poor-quality software costs U.S. businesses $59.9 billion annually, according to a 2002 report from the U.S. Commerce Department's National Institute of Science and Technology (NIST). The NIST study looked not just at the cost of finding and fixing software problems, but also at costs incurred from lost retail transactions and manufacturing product delays.

Those losses are likely to mount as complex software programs are tied together across networks. Think of all the various pieces of corporate data that come together in systems for customer-relationship management, supply-chain management, or enterprise resource-planning—there could be a hundred places where ERP software touches another corporate system, according to Irina Carrel, a senior manager at Mercury Interactive, a company that provides software-testing and -monitoring tools for corporations. And, because of previous bugs, computer-program anomalies or other factors, it's impossible to predict what exactly will happen when two pieces of code come into contact.

"Software is the most complicated thing that the human mind can come up with and build," says Gary McGraw, the chief technology officer at Cigital, a consultancy specializing in improving software quality. "Perfection is unobtainable." (See Dossier, p. 54.)

The medical-device software market is becoming a particular area of concern. The FDA says about half of the 10,000 medical devices on the U.S. market are software-driven—everything from pacemakers to infusion pumps to radiation-therapy machines. FDA watchers say many of the companies developing medical-device software are small. Because of the amount of research-and-development money that goes into medical devices, companies are under pressure to get products out the door.

"We will see more problems" in the medical-device field, says Alan Kusinitz, managing partner of SoftwareCPR, a consulting company that specializes in medical-device software. One of his biggest worries is the ever-increasing number of networked medical devices. Independently, software might function normally, but when connected to code in other machines, it may act unpredictably.

"It's the abnormal stuff that always shows up later in weird circumstances," Kusinitz says. "That's most often where safety problems occur."

There are defense and industrial efforts underway by organizations such as the Sustainable Computing Consortium (SCC) and the Software Engineering Institute, both located at Carnegie Mellon University, to foster programs and standards to reduce software defects. There are also organizations in the healthcare industry, such as the Association for the Advancement of Medical Instrumentation, that are trying to establish standards for software used in medical devices. In addition, new testing tools and services, such as Software Development Technologies' ReviewPro, which examine not just the code but the methodology behind the code, are starting to offer software professionals assistance in vetting their output, as they create it. Also, code-writing practices such as "agile programming" emphasize breaking big projects into small pieces—and getting early and repeated input from users before proceeding.

RAVAGES OF MISCALCULATIONS

Such efforts, though, are too late to help Victor Garcia or the physicists at the cancer institute in Panama who were trying to use imported software to save patients, not make them suffer.

Perhaps nothing shows the ravages of faulty calculations as clearly as cancer. The patients who were suffering in Panama had cancers of the pelvis. Pelvic organs such as the intestines and kidneys are acutely sensitive to radiation. Before a cancer patient such as Garcia is exposed to radiation, a doctor devises a treatment plan that determines what dose of radiation can safely be directed at the tumor. The physician considers the tumor's position and depth in the body, the likelihood that the cancer has spread to surrounding tissue, the location and sensitivity of nearby organs and the best angles of attack.

As part of the plan, the doctor figures out how to place metal shields, known as "blocks," above the area where the tumor is located. These blocks, usually made of lead or a metal alloy called cerrobend, protect normal or sensitive tissue from the gamma rays to come.

The doctor hands his plan to a medical physicist, who feeds information on the size, shape and location of the blocks into a software package. These packages generally create a 3-D picture of how the dose will be distributed, showing how the radiation will "sum" as beams coming in from different angles intersect at depth in the patient's tissue. Once the doctor prescribes a dosage, the software calculates the duration of treatment.

The physicists in Panama were carrying out a doctor's instruction to be more protective, adding a fifth block to the four the hospital often used on patients in cancer treatments. The extra block could help protect patients whose tissues were especially sensitive due to previous surgeries or radiation treatments.

Multidata's planning software was designed to calculate treatments when there were four or fewer blocks, according to the company's general business manager, Mick Conley. Saldaña, however, read Multidata's manual and concluded she could make the software account for a fifth block.

According to an August 2001 report from the IAEA, Saldaña found the software didn't only work if she entered the dimensions of each block individually, up to four. She found it also allowed her to enter the dimensions of all five blocks as a single, composite shape—for instance, a rectangle with one triangular block sitting in each corner and a fifth square block protruding, tooth-like, down into the rectangle from the top.

So, using the mouse attached to her computer, she entered on the screen the coordinates of the specially shaped block—first the inner perimeter of the shape and then the outer perimeter. This is when she felt she was "home free."

After all, when Saldaña entered the data for this unusual-looking block, the system produced a diagram that appeared to confirm its dimensions. She seemed to be getting confirmation from the system itself that her approach was acceptable.

But inside the software, the calculations of appropriate dosages were going awry. The treatment time would be close to correct if Saldaña entered the data for the inner perimeter of the shape going in one direction, say clockwise, and the outer perimeter in the opposite direction, according to the IAEA report. But if she entered the data for the inner and outer perimeters going in the same direction, so that the two loops defining the perimeters crossed, the software essentially locked up—it was not able to accurately recognize the shape and, as a result, miscalculated the treatment times, the report said.

Depending on how many treatments the patients received, they accumulated overdoses ranging from 20% more radiation than was prescribed to a double dose of the potentially harmful rays, the IAEA found.

Inspectors from the FDA were dispatched to Multidata's offices after the agency received reports of patient "radiation overexposures." The inspection ran from May 31 to Sept. 21, 2001. A summary of their findings echoed the IAEA report: "The treatment-planning system miscalculated the dose each patient was to receive due to failure of the software to correctly handle certain types of blocks.... This resulted in a much higher dose being calculated for each patient."

Multidata's Conley says the FDA's finding "is wrong." He says that if you read FDA reports, "you find out the FDA isn't always right.

"Given [the input] that was given," he says, "our system calculated the correct amount, the correct dose. It was an unexpected result. And, if [the staff in Panama] had checked, they would have found an unexpected result."

Conley insists his company has done nothing wrong. He says the physicists at the National Cancer Institute never called Multidata asking for advice or support.

The physicists admit they did not *always* verify the results of the software's calculations, which Multidata's manual said was "the responsibility of the user." Saldaña says the hospital was treating more than 100 patients per day using the one Cobalt-60 machine. The IAEA also found that whatever steps the hospital took to ensure the radiation machine was operating properly only addressed the hardware. There was no quality-assurance program for the software—or its results. In the day-to-day operations of the cancer institute, that meant the physicists were not required to tell anyone they had changed the way they entered data into the cancer-therapy system. As a result, no one on staff questioned the software's results.

Had the hospital verified the dosages—by manually checking the software's calculations; or by testing the dosages in water before radiating patients, a procedure that Conley argues is standard medical practice in much of the rest of the world—the staff would have caught the overdoses in time to avoid harming anyone.

But independent experts not associated with the case say software that controls medical equipment and other life-critical devices should be designed to pause or shut down if told to execute a task it's not programmed to perform. "If a computer can make a user kill people, it's like a loaded gun," says Jack Ganssle, an engineer whose Ganssle Group advises companies and developers on how to create high-quality software. "A user shouldn't be able to do anything that causes a machine to be dangerous."

But the Multidata software continued to operate.

CAUSE OF DEATH

As tragic as it is, the Panama incident does not stand alone. In all, *Baseline* has found no fewer than a half-dozen cases in which software has contributed to loss of life. (See chart, p. 47.)

At least three deaths were blamed on a software glitch that crippled the East Coast's power grid last summer. In 1997, the safe-altitude warning system at Guam International Airport inexplicably generated an excessive number of false alarms that planes were flying too low. As a result, air-traffic controllers cut back the distance scanned by the system from 54 miles to 1 mile. The change prevented controllers from warning pilots of Korean Airlines Flight 801 that they were flying toward a mountain. The crash killed 225. There are also scores of personal injuries in which software was at least partly to blame. A rider on a gyroscopically controlled Segway scooter suffered a head injury because of a software-design gap and, according to the National Highway Traffic Safety Administration, more than 476 people have been hurt because of a problem with a General Motors antilock braking system in use from 1991 to 1996. GM said the braking system wasn't designed to check for certain drive-train variables.

Certainly, deaths and injuries that can be in some fashion tied to software are statistically rare. Overall, software quality is "generally pretty good," says James Gosling, a Sun Microsystems vice president. But Gosling, regarded as the father of the Java programming language, which can be used to build applications that can run across diverse computers, says code-writing in many cases is still flawed.

Many specifications and designs aren't thought out well enough. Programmers, no matter how good, make logical mistakes. In addition, testing procedures often aren't rigorous enough, he says. And today, with so many software programs interacting with other software programs, there's no way to predict what will happen when two pieces of code come in contact with each other for the first time.

"The quality fight is never-ending," he says.

The threat of physical harm and crippled lives is escalating, now that software drives not just healthcare machinery, but our cars and our household appliances as well. It runs elevators and amusement-park rides. It controls just about every manufacturing plant, utility and business office in the country. As software becomes more pervasive, software quality—long a discussion confined to software-development circles—becomes an issue for business executives, product managers, factory floor supervisors and, as the physicists in Panama found out, anyone who uses software in the workplace.

PROJECT PLANNER
IF YOU WANT TO START PREVENTING SOFTWARE ERRORS, HERE'S HOW (SEE FOLDOUT).

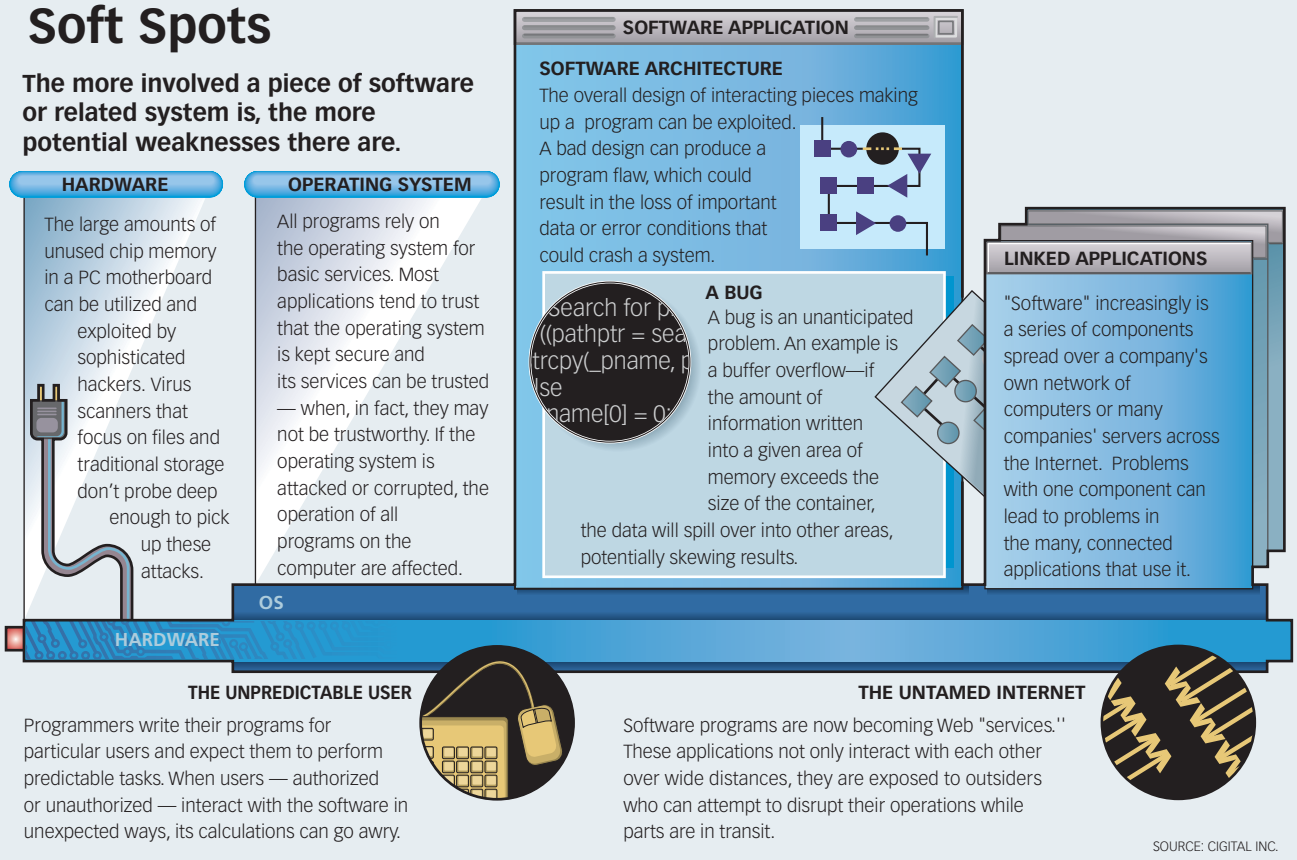
"What can you do today without software?" asks Pradeep Khosla, head of the department of electrical and computer engineering at Carnegie Mellon University. "Nothing."

But all software has bugs.

For every thousand lines of code developed by commercial software makers or corporate programmers there could be as

Soft Spots

The more involved a piece of software or related system is, the more potential weaknesses there are.



THE UNPREDICTABLE USER

Programmers write their programs for particular users and expect them to perform predictable tasks. When users — authorized or unauthorized — interact with the software in unexpected ways, its calculations can go awry.

THE UNTAMED INTERNET

Software programs are now becoming Web "services." These applications not only interact with each other over wide distances, they are exposed to outsiders who can attempt to disrupt their operations while parts are in transit.

SOURCE: CIGITAL INC.

INFOGRAPHIC BY MIKA GRÖNDAHL

SOFTWARE QUALITY—BY THE NUMBERS

There can be as many as **20 to 30 bugs per 1,000 lines of software code.**
—Sustainable Computing Consortium

There are **no methods** of removing software defects or errors **that are 100% effective.**
—"Software Quality: Analysis and Guidelines for Success," by Capers Jones

32% of organizations say that they release software **WITH TOO MANY DEFECTS.** —Cutter Consortium

38% of organizations believe they **LACK** an adequate **SOFTWARE QUALITY ASSURANCE PROGRAM.** —Cutter Consortium

27% of organizations **DO NOT conduct any formal QUALITY REVIEWS.** —Cutter Consortium

Formal design and code inspections average about **65% in defect removal efficiency**. MOST FORMS OF TESTING ARE LESS THAN **30% EFFICIENT.**
—"Software Quality: Analysis and Guidelines for Success," by Capers Jones

Developers spend about **80%** of DEVELOPMENT COSTS on IDENTIFYING AND CORRECTING DEFECTS. —The National Institute of Standards and Technology

Peer reviews of software WILL CATCH 60% OF DEFECTS.
—Institute of Electrical and Electronics Engineers

many as 20 to 30 bugs, according to William Guttman, the director of the SCC, a group of businesses and academic institutions looking for ways to make software more dependable. Many common programs have a million or more lines of code. Sun says its Solaris operating system has more than 10 million lines of code. Even a high-end cell phone can have 1 million.

In a one-million-line piece of code, even if you only have one bug per thousand lines, you're still going to have 1,000 bugs, says Michael Sowers, executive vice president at Software Development Technologies, a software-testing company.

In today's software, says Khosla, "you have to assume there are some bugs in the code."

Just look back at the first major case of code that killed, in healthcare.

The Therac-25 was one of the first "dual-mode" radiation-therapy machines, which meant that it could deliver both electron and photon treatments. Electrons are used to radiate surface areas of the body to kill cancer cells. A photon beam, normally called an X-ray, can be a hundred times more powerful and as a result is used to deliver cancer-killing radiation treatments deeper into the body. According to Prof. Leveson's account, the machine was "more compact, more versatile, and arguably easier to use" than its predecessor machine.

But, according to Prof. Leveson's 1995 book "Safeware" and other accounts, there were a number of flaws in the software that led to the Therac-25 radiation overdoses at health facilities in Marietta, Ga.; Tyler, Texas; Yakima, Wash.; and elsewhere. In all, three people died.

One of the problems manifested itself in 1986 when a physicist tried to change machine set-up data—such as radiation dosage and treatment time—that had been keyed into the software.

The machine went through a series of steps to set itself up to deliver either electrons or photons and the dosage of the selected beam. As data was given, the machine recorded the information and then followed the instructions.

In some cases, however, operators realized while setting up the machine that they had entered an incorrect piece of information. This could be as simple as unintentionally typing in an "X" for an X-ray (or photon) treatment instead of an "E" for an electron treatment.

In "fixing" that designation, an operator would move the cursor up to the "treatment mode" line and type in an "E." The monitor displayed the new entry, seemingly telling the operator that the change was made.

But in the case of the Therac-25, the software did not accept any changes while going through its eight-second-long set-up sequence. No matter what the screen might show, the software grabbed only the first entry. The second would be ignored.

Unaware the changes did not register, operators turned on the beams and delivered X-rays, when they thought they were delivering electrons. According to Leveson's account, patients received such incredibly high quantities of radiation that the beams burned their bodies. Patients who should have received anywhere from 100 to 200 rads of radiation were hit instead with 10,000 to 15,000 rads, in just one or two seconds. A thousand rads is a lethal dose.

The Therac-25, according to John Murray, head of software regulatory efforts at the FDA, was a "seminal event" for the agency. After the incident, the FDA for the first time turned its attention to the software that had begun to control medical devices.

The FDA has the power to inspect the work of manufacturers; to ask manufacturers to recall products; to have federal marshals seize products if a voluntary recall isn't done; and to ask the courts to issue injunctions against the distribution of products if a manufacturer does not have good manufacturing procedures in place.

To help software manufacturers, the FDA issues "guidance" documents that recommend that manufacturers follow generally accepted software-development standards; keep track of their design specifications; and conduct formal reviews and tests of the code they produce. Arne Roestel, Multidata's president, says the company followed the FDA recommendations.

But there are few specifics. According to the FDA's "General Principles of Software Validation," which went into effect in January 2002, "This guidance recommends an integration of software lifecycle management and risk management activities. Based on the intended use and the safety risk associated with the software to be developed, the software developer should determine the specific approach, the combination of techniques to be used, and the level of effort to be applied."

In the wake of Panama, some industry experts wonder if there's enough oversight of medical-device software—or, for that matter, software development in general. They say the time might be right for tougher regulation.

Software engineer Ganssle, for one, notes that programmers don't need any form of certification or license to work on commercial software, including life-critical medical device software. Yet, he says, "In Maryland, where I live, if you want to cut hair, you need to be licensed."

Besides the FDA, there are few federal agencies policing software-development practices. The Federal Aviation Administration oversees the flight-control software in commercial aircraft. The Nuclear Regulatory Commission (NRC) watches over the software that runs nuclear plants. And that's about it, for oversight of commercial software. The Occupational Safety and Health Administration, the Consumer Product Safety Board and other agencies charged with protecting factory workers, professionals and consumers say they don't worry about the quality of software in tools or toys.

WHAT WENT WRONG

Overlooking the skyscrapers of downtown Panama City, amid towering palm trees and gracious homes in the old Canal Zone, sits Gorgas Hospital, an imposing concrete structure which now houses Panama's National Cancer Institute. This is a public hospital. No Panamanian is turned away.

On a Monday morning in January, at least 50 patients and their family members, including Victor Garcia and his wife, are visiting the institute. The patients walk slowly up the driveway; sit quietly on the patio under the lush vegetation that surrounds the building; stand in the lobby. They are all waiting for treatment.

This is not even the hospital's busiest day of the week. That is Tuesday, when the clinic offers every citizen, even those without a doctor's referral, free diagnoses of the skin cancer that tends to flourish under the equatorial sun.

Cancer is a leading cause of death in Panama: prostate cancer for men, endometrial and cervical cancer for women. And those are unlikely to be just the sun's fault. Many Pan-

amanians blame the United States' testing of the chemical defoliant Agent Orange in the Canal Zone during the Vietnam War. Since 1997 the number of new cancer patients in Panama has more than quadrupled, according to the cancer institute. The hospital now sees 10 to 15 new patients per day and performs 300 cancer surgeries per month.

The victims of the faulty radiation treatments in 2000 and 2001 span the breadth of Panamanian society. Among the dead are Margarita Sevillano, a folksinger; Walter Chandler, a professor at the University of Panama; and Rosa Vergara, a nun. Many of the dead lived in the barrios in the hills above downtown, where chickens peck along the roads, laundry flaps from porches and brightly painted stucco houses are interspersed with small shops and Internet cafés.

The hospital's radiotherapy unit is critical to Panama. When the IAEA's investigation, in May 2001, slowed the hospital's routine, patients lined up waiting to be treated. That led the Panamanian ambassador in Austria, Jorge Perez, to urge the Vienna-based agency to hurry up. "Those who could afford to went to the private clinics," Perez says. "Those who could not, waited."

The difference in cost to the Panamanian government is stark. Garcia's treatment, for example, which cost him virtually nothing at the National Cancer Institute, would cost $4,000 at a private hospital, using a Cobalt-60 machine. Using a higher-powered, more-precise linear accelerator, the bill would escalate to $10,000.

The current chief of the cancer institute's radiotherapy unit, Dr. España de la Rosa, asserts that some patients died in 2001 waiting for treatment, while the overdoses were being investigated. She says she does not know how many.

But the survivors of the overdoses didn't fare well, either. The governments of France and Argentina each offered to take two of the over-radiated patients and treat them for a year at no charge. Panama sent no one. "We are a small country, and everybody knows everybody," Ambassador Perez says. "How do you decide who to send?"

The overdoses occurred not in the newly renovated Gorgas Hospital on the hill, but in the cramped Justo Arosemena Avenue facility downtown, which the hospital was in the process of vacating. The Multidata software and the Cobalt-60 teletherapy machine manufactured by Theratronics had been installed there in 1993. According to a letter written to Multidata by ProMed, the Panamanian distributor that sold the hardware and software, the hospital was looking for cheaper software because it couldn't afford the software that Theratronics typically supplied with its radiation machine.

ProMed services manager Camilo Jorge says he doesn't remember the price difference, but he knows the hospital never purchased a maintenance contract for the software—only for the radiation machine. By 1997, hospital staff was so concerned about the possibility of unintended excess exposure that they warned in a report requested by the Ministry of Health of "overexposure of radiation-therapy patients due to human error" unless conditions at the hospital improved.

In the report, the staff claimed the hospital was understaffed and poorly equipped, and they asked for more frequent maintenance on the Cobalt-60 teletherapy machine. The contention: the machine was being used 3,780 hours per year, nearly twice what the maintenance program recommended.

EIGHT FATAL ACCIDENTS

Mishaps in which software-related problems were reported to have played a role.

Date	Deaths	Detail
2003	3	Software failure contributes to **power outage** across the Northeastern U.S. and Canada.
2001	5	Panamanian cancer patients die following **overdoses of radiation**, amounts of which were determined by faulty use of software.
2000	4	**Crash of a Marine Corps Osprey tilt-rotor aircraft** partially blamed on "software anomaly."
1997	225	Radar that could have prevented **Korean jet crash** hobbled by software problem.
1997	1	Software-logic error causes infusion pump to deliver **lethal dose of morphine sulfate.** Gish Biomedical reprograms devices.
1995	159	American Airlines jet, descending into Cali, Colombia, **crashes into a mountain.** Jury holds maker of flight-management system 17% responsible. A report by the University of Bielefeld in Germany found that the software presented insufficient and conflicting information to the pilots, who got lost.
1991	28	Software problem prevents Patriot missile battery from picking up **SCUD missile**, which **hits U.S. Army barracks** in Saudi Arabia.
1985	3	Software-design flaws in Therac-25 treatment machine lead to **radiation overdoses** in U.S. and Canadian patients.

The staff also asked the hospital to have Multidata do "preventive maintenance" on the software. But the software was never maintained, and by 2000, the hospital was using just the one Cobalt-60 machine to treat all patients, according to Saldaña. A second, older machine was retired.

By then, two of the hospital's five radiation physicists had quit. Saldaña says the remaining three did the work of five, which sometimes required 16-hour days.

Victor Garcia remembers waiting five to six hours for every treatment. And after each of those six treatments, he felt sicker. As his intestines struggled to slough off cells killed by the radiation, he developed diarrhea. Burns seared through the flesh on his back. He lost 30 pounds. Hospital doctors told him the symptoms were normal. One reason it took hospital staff seven months to discover the overdoses, according to hospital director Juan Pablo Bares, is that patients with pelvic cancers often show symptoms of radiation toxicity, and the number of patients overdosed was small compared to the number being treated.

But another reason was the complexity of the software. The glitch involving Multidata was activated only under very specific circumstances—when the dimensions of the blocks that defined the patient's treatment area were entered in a particular way. If the blocks were treated as a single, composite shape, and the descriptions of their dimensions were entered so that the "loops" that defined the inner and outer perimeters of that shape crossed, the software would increase patients' treatment time, the IAEA report said.

As patients began to sicken and then die, the staff hunted for the cause. Saldaña remembers that by March 2001, she was thinking the problem had to be the software. But even then she discovered it by accident: On the morning of March 2, according to a statement she gave to the prosecutor's office, she was calculating dosages for two patients with equivalent treatment areas and treatment depths and suddenly realized that the treatment times that came out of the software weren't even close.

And so began the hospital's effort to unearth the causes of the overdoses.

Radiotherapy expert J. Francisco Aguirre, who investigated the overdoses for the Panamanian government as part of a team from the M.D. Anderson Cancer Center in Houston, says the calculation error was a problem that occurred with algorithms in older software used to plan treatments, a linkage that Multidata president Arne Roestel denies. Aguirre says the error was so obscure he wouldn't have thought to look for it—except that while he was in Panama, he remembered seeing a physicist in the U.S. cause a similar error 10 years before.

"The trick is how to tell the computer what are [empty] holes and what is solid," Aguirre says. "If the lines you are digitizing cross along the way, you fool the computer."

Indeed, during the IAEA's May 2001 investigation, the agency found ways to get the software to miscalculate treatment times that the hospital staff hadn't tried. Investigators were able to enter the dimensions for one block, two blocks or four blocks of varying shapes, and every time they treated them as a single block and entered the coordinates so that the perimeter loops crossed, the software always increased the treatment times.

Both the M.D. Anderson and IAEA investigating teams found Multidata's manual hard to understand. "It does not describe precisely how to digitize co-ordinates of shielding blocks and there are not enough relevant illustrations," the IAEA report said. "In addition, it does not provide specific warning against data entry approaches that are different from the one described."

The Houston team's report said: "The manufacturer's manual of instructions was reviewed, and no indication was found in the instructions on how to digitize the blocks, or procedures to avoid, that could result in bad calculations."

On Aug. 10, 2001, in an "urgent notice" to users, Multidata used a series of diagrams to describe how the "crossing-loop" problem—which the company described as a "data entry sequence that creates a self-intersecting shape outline"—would not be acceptable to its program and would cause miscalculations. And it appeared to specifically absolve those users who, like Saldaña, had tried to get the software to give results when five shields were being placed on patients instead of four.

"Digitizing direction and exceeding the number of blocks, numbers of points per block or the block shape have no unexpected effect on the dose calculation," the notice said.

THE INVESTIGATIONS

Multidata's Mick Conley, in an interview with *Baseline* on Feb. 19, maintained that his company first heard full details of the overdoses from the FDA and the NRC in June 2001.

But Aguirre says that is not true. His team notified Multidata in April, he says, two days after they returned from Panama to the U.S.

"We told them, 'Your equipment has had an accident, go and make sure that for all the systems you've sold in the world, people are aware this is a problem,' " Aguirre says.

The IAEA report confirms that the Houston team notified Multidata in April, "and that it was impressed upon Multidata to send someone to Panama as soon as possible to resolve this problem."

But Roestel says Multidata could not get enough information from the hospital or the Houston team on exactly what had happened.

In addition, Conley says, in order for the company to send representatives to Panama, the hospital would have needed a service contract with the company, which it didn't have, although he adds that Multidata provides telephone support to anybody with its product.

But before sending a team out, Conley says, "you really have to have an idea of what the problem is." If another customer called in a similar situation, he says, Multidata would want a clear explanation of what the problem was so that they "knew what kind of person to send to a place and what kind of tools to send along with them."

On May 18, after several patients had already died, the Panamanian government announced the overdoses, and international agencies began to act. In the U.S., the NRC sent out warnings on June 1 and again on June 6 alerting hospitals licensed to perform radiation therapy of the overdoses—and that Multidata's software was involved.

Multidata issued its first warning to customers on June 22, although it did not say which versions of its software were affected or how the overdoses happened. The company directed users to "follow instructions in the user manual...follow a written quality assurance procedure...and perform verification measures."

Multidata issued what Conley calls a "voluntary in-field correction," starting in the fall of 2001. The software patch checked treatment-planning calculations and rejected anything that was not identified by the system as a valid shape. "We changed the software so that [the Panama incident] could not happen again," he says.

Conley maintains the company was initially unaware the software was still being used in Panama. "We never heard from [the hospital]," he says. Conley and Roestel also contend the software was fine and that the problem was user error.

The IAEA report did note that quick action by the hospital staff could have prevented the overdoses.

At the end of May 2001, right after the FDA became aware of the accidents in Panama, it sent its examiners to inspect Multidata. The FDA found that Multidata had received at least six complaints about "calculation errors related to the failure of the firm's radiation treatment planning software to correctly handle certain types of blocks (polygons)."

The report said: "As of 6/1/01, there was no documentation that any complaint or incident report analysis had been performed, or corrective action developed or implemented [by Multidata]. In addition, the firm had been aware of this failure since at least 9/92."

Separately, the NRC published the findings of the IAEA in an Information Notice to operators of radiotherapy machines dated Nov. 20, 2001. The notice said: "Specifically, the staff modified its procedures and entered data for multiple shielding blocks together ('digitized' the blocks), as if they were a single block. The data were accepted by the treatment planning system, but the [Multidata] software calculated incorrect treatment times. Using incorrect treatment times resulted in significant radiation overexposure to patients."

In 2003, Multidata signed a consent decree with the FDA that precludes the company from making or selling software for radiation-therapy devices in the U.S., although it can still export its products. According to the FDA's injunction announcement, Multidata failed to meet the FDA's manufacturing practices and design standards.

Of all the criticism the firm received, perhaps the harshest was a statement made by FDA Commissioner Mark McClellan when the injunction was made public on May 7, 2003.

"Multidata Systems has a nine-year history of violations and failure to correct them," he said. "Despite repeated warnings, the company continued to manufacture its medical devices in a way which put the public health at risk."

Indeed, the FDA has taken Multidata to task several times over the last 10 years for its software-development practices.

The FDA's policy is to inspect any company that makes medical devices about every two or three years. Each time its examiners visited Multidata they found problems. The FDA would not say how common it is to find problems on each company visit, but Gladys Rodriguez, head of the FDA enforcement unit that deals with medical devices and radiological health products, says an injunction is a "rare" action.

According to documents obtained by *Baseline* under the Freedom of Information Act, the FDA inspected Multidata's products and manufacturing operations four times in the past dozen years.

In 1993, 1995 and 1998, FDA inspectors found the same deficiencies in Multidata's software-development process coming up time and again: a lack of good software specification and documentation procedures to guide and control the software-development and change process; insufficient documentation to show that the software had been properly tested to see if it worked; and inadequate investigation into customer complaints.

The FDA inspectors' report from the 1993 investigation noted: "[C]omplaint review found a number of reports of software errors or 'bugs' which indicate Multidata's software testing is incomplete. Several of these complaints reported incorrect dose calculation described as 'off by about 20%', and 'bizarre' and 'dramatic.' Follow-up investigation of these and other complaints found many software errors present in software shipped to customers that could have been found with structured, thorough, and rigorous testing throughout the software development process using basic software analysis and testing techniques."

'You have to assume there are some bugs in the code.'

—PRADEEP KHOSLA, HEAD, ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT, CARNEGIE MELLON UNIVERSITY

Conley says the company looked into the complaints and corrected any problems. "We fixed what we found," he says, adding that some of the complaints the company looked into were not software-related, but resulted from users being unfamiliar with its product. And, he says, there's no link between the 1993 report and the accidents in Panama.

After the FDA's 1998 inspection, the agency sent a warning letter to the company, outlining the findings from its review and instructing the company to look into its software-development process and correct the deficiencies noted in its report. The agency told Multidata that if the company didn't take care of the problems, the FDA would be forced to take further action, which could include fines and an injunction against the company.

The FDA also told Multidata it wanted the company to notify the agency within 15 days of the steps the vendor was going to take to address the problems.

According to the subsequent FDA reports, Multidata never responded to the 1998 letter.

"We usually get action from our warning letters," Rodriguez says. Multidata's Roestel admits that the company did not respond to the FDA's letter.

In 2001, after the deaths in Panama, FDA inspectors found many of the same problems they found earlier. Multidata, it said, had no mechanisms for addressing incomplete or ambiguous software requirements, customer complaints were not being properly recorded, and there was no comprehensive testing plan to demonstrate that its software was "fit for use."

Conley says the company has been working hard to rectify the complaints identified in the FDA's 2001 inspection. He says Multidata, in fact, was looking for ways to better track software fixes and problems, including the installation of a computerized system to record and store customer complaints, when the FDA issued its injunction.

Conley admits Multidata didn't address issues quickly enough for the FDA.

"We're a small company that didn't always react in a timely fashion," he says. "We did what we were supposed to do, [but] we didn't file the proper reports for it."

To have the injunction lifted, the FDA says Multidata must improve its design and manufacturing methods; upgrade its record-keeping mechanisms; and retain a medical-device design expert to inspect the company's manufacturing activities, check over its software code, and report back to the FDA. The outside expert must have no financial ties to the company other than the consulting agreement for this series of tasks.

Conley said in February 2004 that Multidata will have its development practices reviewed by Bio-Reg Associates Inc., which describes itself as a "regulatory consulting firm conveniently located close to the FDA in Washington, D.C."

"They are staking their reputation on the line by representing us to the FDA," Conley says. "They would not take us if we were a shlock outfit."

EXAMINING THE REGULATORS

Despite the strong action it took against Multidata, FDA watchers say the agency—one of the few empowered to regulate software in any fashion—still does not go far enough to insure the integrity of the computer programs that are the brains of medical devices.

"Thinking the FDA is some sort of watchdog group is an exaggeration," says Bob Morton, a software-quality expert and a former head of the FDA unit in charge of radiation-therapy equipment.

The FDA approves medical devices under one of two mechanisms: premarket approval or premarket notification.

Premarket approval is reserved for technologies that are radically different from anything on the market, such as a breakthrough pacemaker product. These products are run through scientific reviews and tests to ensure that they are both safe and effective. Products which fit into an existing category of devices are subject only to the premarket-notification process, under which the FDA neither tests products nor requires a manufacturer to conduct its own field trials.

Multidata won approval for its software in March 1997 under the notification process, since a number of radiation-treatment planning packages were already on the market.

In this procedure, a manufacturer such as Multidata submits paperwork that details how they designed, produced and tested the new product. Agency officials then pore over the documents looking for potential flaws, problems in testing or other likely trouble. If everything is in order, the FDA gives the company clearance to sell its product.

Last year, 3,500 of the 4,000 medical devices approved by the FDA came through the notification process.

"Can we rely on the FDA to police the medical-device industry in a very reliable way, via the premarket submissions and the inspections?" asks SoftwareCPR's Kusinitz. "No. I think we're primarily dependent on the...good intentions of the medical-device manufacturers."

In their defense, FDA officials say it's not feasible to test every product. "We regulate 10,000 different types of products. How do you come up with tests for 10,000 products?" says Murray, the FDA software expert. Besides, he says, the FDA has no conclusive data that outside testing would improve the quality of software anyway.

DOSSIER: CIGITAL

BUG ZAPPERS

Cigital chief executive officer Jeffery Payne likes to deliver good news to his customers first, when possible: that their systems are 100% secure. But, sometimes, he has to deliver bad news as well. That the software is totally unreliable.

"People are worried about security, but in the end the problems with software are the age-old problem, that it just doesn't work," he says.

Cigital, a software-consulting firm in Dulles, Va., has thrived by figuring out exactly why computer code doesn't behave the way it's supposed to. The firm, which markets itself as a provider of "software-quality management" services, operates as a kind of forensic-analysis squad for software developers. **"They're one of the elite companies doing anything like this," says Avi Rubin, an associate professor of computer science at Johns Hopkins University.**

Payne and Cigital's other founder, Jeffrey Voas, first met as graduate students in the computer science department at The College of William and Mary in the late 1980s. The duo (referred to inside the company as "the two Jeffs") met up again in 1990, as Voas was finishing his doctoral thesis about how to make software more reliable. "He was talking about the fact that software didn't work very well, and it started me thinking there was a business opportunity there," Payne says.

There was—but for Cigital, it's been a relatively small one. **The 70-person company, which works on between five and 15 projects at a time, expects to pull in somewhere between $10 million and $20 million in revenue this year, Payne says.**

That's petty cash for big technology-consulting outfits

THE COMPANY

HEADQUARTERS: 21351 Ridgetop Circle, Suite 400, Dulles, VA 20166

PHONE: (703) 404-9293

URL: www.cigital.com

TICKER: Private

EMPLOYEES: 70

BUSINESS: Provides consulting services to analyze and fix errors in software, and to improve software-development processes.

FOUNDED: 1992, as Reliable Software Technologies.

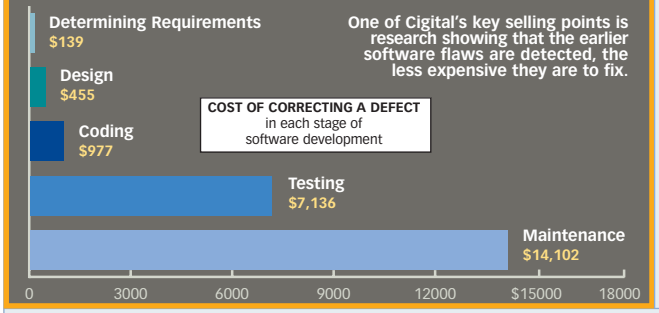
EXECUTIVES: Jeffery Payne, president and CEO; Jeffrey Voas, chief scientist; Gary McGraw, chief technology officer.

FUNDING: Raised $4 million in August 2002 from Mid-Atlantic Venture Funds, Blue Water Capital and the Washington Dinner Club LLC. Co-founders Payne and Voas together retain the majority equity stake.

FINANCIALS: Revenue of $8.3 million in 2002, according to Hoovers; the company says it's profitable.

COMPETITORS: Accenture; @stake; BearingPoint; EDS; IBM Global Services; PricewaterhouseCoopers.

THE PRICE OF BUGGY CODE



SOURCE: IEEE COMPUTER SOCIETY, JANUARY 2001

such as EDS or IBM Global Services. But Cigital doesn't aspire to vastness, although it sees great growth in companies' needs for improving the reliability and security of software. "We're not going to grow to be an Accenture, and we don't think we need to," Payne says. "We're focused. This is all we do. You build a business by finding that one thing you do better than anyone else."

Considering the small patch of green it putts on, Cigital has plenty of fans. It has worked with such companies as General Electric, Nortel Networks, Pfizer, Raytheon, Texas Instruments, MasterCard and Visa, as well as several government agencies, including the Air Force Research Laboratory, the Defense Advanced Research Projects Agency and the National Security Agency. Payne says Cigital has even done some consulting for Microsoft, though he declines to elaborate. (In February 2002, Cigital publicized a security flaw it discovered in the Redmond company's just-released C++ programming tools, but a Cigital spokesman says this was completely unrelated to any work it has done for Microsoft.)

Cigital's CEO, Jeffery Payne

For MasterCard, Cigital's testing and analysis of Java-based smart cards reassure the credit-card company that it has covered its bases. "I don't like seeing articles in *The New York Times* that cause my CEO to call me up and say, 'Are our smart cards vulnerable to this?'" says Simon Pugh, vice president of infrastructure and standards at MasterCard. "Now I'm in a position to say, 'We've known about this problem for three years, and we have accounted for it in our testing procedures.'"

Pugh adds that **Cigital is very straightforward about discussing any flaws it finds. "They don't overreact or oversell," he says.** "It's very easy to get melodramatic about security issues.

NASD, the private organization that regulates the securities industry, started working with Cigital in 2001 to automate its testing processes. "These guys had top-notch people," says Martin Colburn, NASD's chief technology officer, who oversees a team of 100 programmers. **"They had great skills not only in developing the testing, but they also have an art form around using the standard tools that are out there."**

Meanwhile, human-resources consultancy Towers Perrin last fall hired Cigital for three weeks to audit how its retirement-systems group tested applications. "They delivered exactly what was expected, on time and on budget," says Greg Velott, manager of retirement systems at Towers Perrin. Based on Cigital's recommendations, Velott and his team were able to identify a higher percentage of critical bugs earlier: Fewer than 5% cropped up in the last two weeks of their latest six-month release cycle. In previous cycles, the group had been finding more than 25% of all major application defects in that final period.

Cigital's Payne predicts that more companies will come to realize how closely software is tied to the bottom line. "In 1992, I couldn't sell this to my mom," he says. "It wasn't critical enough to get the attention of business executives. Now we don't have to convince people that software quality is a problem. Now they're looking for an answer." —TODD SPANGLER

THE TECHNOLOGY

CIGITAL'S STOCK-IN-TRADE IS NOT software. It's a philosophy about how software ought to be built.

The cornerstone of the company's approach is the principle that reliability, security and performance must be designed into software from the beginning. Then, software must be tested at every stage of the development process. This methodology helps programmers detect errors earlier—thereby reducing the cost of correcting them—and produce more-robust software.

Cigital also uses risk-management techniques to determine the business consequences of software failures, because testing even a small program exhaustively would take hundreds of years.

Such ideas may seem obvious now. But when Cigital was founded 12 years ago, the concept of iterative quality testing was foreign in the commercial-software world. "Cigital was one of the first firms to have people on staff who understood the issues of why software fails," says Theresa Lanowitz, an analyst at Gartner Inc.

Experts say Cigital has achieved several technical breakthroughs. Its automated testing tool, for example, intentionally introduces errors in order to predict how software will fail. Robert Stoddard, a senior software engineer at Motorola, estimates that 85% of the bugs the mobile-phone group ended up finding would have gone undetected without the Cigital application. "This went after a whole class of errors we weren't even testing for," he says. —T.S.

REFERENCE CHECKS

NASD
Martin Colburn
CTO
(202) 728-8000
Project: Financial regulatory-services organization used Cigital tools and methodologies to automate the testing of several applications, including its central registration system.

MOTOROLA
Robert Stoddard
Distinguished Member, Technical Staff
robert.stoddard@motorola.com
Project: Mobile-phone manufacturer used a customized version of Cigital's testing software to improve the reliability of its products' underlying code.

TOWERS PERRIN
Greg Velott
Manager, Retirement Systems
(215) 246-6000
Project: Human-resources consulting firm hired Cigital to audit the processes used in testing the applications that run its retirement services line of business.

RAYTHEON
Gavin T. Watt
Senior Principal Systems Engineer
gwatt@raytheon.com
Project: Aerospace company's Navigation and Landing Systems division used Cigital technology to identify errors in its Global Positioning System-based software.

MASTERCARD INTERNATIONAL
Simon Pugh
VP, Infrastructure and Standards
simon_pugh@mastercard.com
Project: Credit-card network engaged Cigital in 2001 to analyze the security of applications running on Java-based smart cards.

SRS TECHNOLOGIES
Trase Travers
VP/General Manager
(321) 784-7828
Project: Contractor enlisted Cigital to assure software quality for systems used at NASA's Goddard Space Flight Center to develop and launch Earth-orbiting satellites.

Critics, however, still take issue with the FDA being so reliant on human review.

"The problem with human review is that it's not infallible," says Jonathan Jacky, a radiation-oncology research scientist now working at Microsoft Research, Microsoft's computer-science research organization. Humans, he says, "just might overlook something."

The FDA admits that it doesn't have the manpower to look at every line of code and "things do get missed," according to Timothy Ulatowski, director of compliance at the FDA unit that oversees medical devices.

Indeed, some bugs are even *allowed* in the software. According to FDA documents, a list of all bugs left in a system must be submitted, plus documentation that those bugs aren't a safety concern. The FDA also asks manufacturers to submit a schedule for when they plan to fix the bugs. None of the bugs, however, can be considered a safety issue. Multidata says it submitted its bug list to the FDA and that it fixed all bugs.

The FDA's task will only get more difficult as time goes on. While the number of medical devices approved has remained steady—roughly 4,000 products a year for each of the last five years—the devices and their software are becoming more pervasive and more complex. Already, about half the medical devices approved for market contain software, and FDA watchers expect that percentage to grow. "Each iteration of [a] device tends to put more software in," says Morton.

And, Morton says, "devices are being released before they're ready." He won't name companies or products, citing confidentiality agreements, "but it's true," he says.

The FDA, in its defense, maintains that it's up to the task. If anyone wants to know how tough the FDA is, says Ulatowski, "ask Multidata."

BACK IN PANAMA

Even though Multidata did send representatives down to Panama to deliver a patch for the hospital's software and provide its staff with additional operating instructions, the hospital had stopped using the software in June.

Patients were still being treated with the Cobalt-60 teletherapy machine, but the physicists calculated the patients' treatment times the old-fashioned way—by hand.

As a result, the hospital could only handle around 60 to 70 patients per day, instead of 100. That led to an even longer waiting list and forced the Panamanian government to start subsidizing private hospitals, where it sends those patients who are employed and therefore covered by social security. So far, says Dr. de la Rosa, the government has spent $10 million subsidizing private treatments.

Meanwhile, the three physicists are free on their own recognizance while they await trial. Two of them—Saldaña and Alvaro Mejia—continue to work at the National Cancer Institute. The physicists are funding their own defense, even though Saldaña, for instance, makes $585 a month.

Saldaña, who has worked at the institute since 1988, says it is difficult to continue after the overdoses, but "if we did not work, the patients would die." Ricardo Lajon, the chief physicist, calls Saldaña "one of the best physicists we have."

The Houston team also praises the physicists. "This was an unfortunate occurrence, which we believe was not foreseeable," its report said. "However once discovered the actions taken were appropriate and the cause was quickly found. The personnel involved are to be commended."

Prosecutor Cristobal Arboleda acknowledges the peculiarity of having hospital workers accused of second-degree murder continue to treat patients. But he says they must be presumed innocent until found guilty and cannot be fired before the trial.

Besides, he says, "administratively, the hospital needs them." Due to a dispute with the Ministry of Health, the entire radiotherapy department has been operating without a license to deliver radiation, a fact that Multidata is using as part of its legal defense. But if the department were shut down, Arboleda says, "90% of the cancer patients in Panama would die."

Saldaña today appears calm for someone who faces the possibility of two to four years in prison. Her mother takes care of her 13-year-old son in a town in the highlands, and that will continue if she goes to jail. The families of two of the dead patients have hired a private prosecutor to pressure the judicial system to convict the physicists, a common practice in Panama. Arboleda expects other civil suits to be filed in Panama depending on the outcome of the criminal trial.

Meanwhile, the families' lawsuits against Multidata and MDS have been dismissed in both countries—for lack of jurisdiction in Panama and for "forum non-conveniens" in the U.S. On Jan. 15, 2004, St. Louis County Circuit Court Judge Emmett M. O'Brien told the families to re-file their suit in Panama, where the overdoses occurred. Yet Judge Zoila Rosa Esquivel of the First Court of Justice of the Civil Circuit in Panama had dismissed the suit on April 30, 2003, saying that the case could not be pursued in two countries at the same time. In effect, by taking the companies to court in St. Louis County first, the families forfeited the right to take them to court in Panama, according to Esquivel's ruling.

Now the families are trying again in Panama. Judge O'Brien said the companies need to be given the chance to respond to the charges in Panamanian court, before the case can be reconsidered in the U.S.

Judge O'Brien's decision is a victory for Multidata and MDS, which fought to get the suit tried in Panama. Judgements awarded in Panama tend to be low, compensating victims just for actual damages, notes Edgardo Molino-Mola, a former Panama Supreme Court Justice. And since Panamanian judges permit both sides to engage in delaying tactics, such as filing motions with no substance, cases may not be resolved for 10 or 15 years.

Regardless of how the court cases end up, some good has come out of the tragedy. The Government of Taiwan donated two new linear accelerators to the National Cancer Institute, to replace its single, aging Cobalt-60 machine, and the Ministry of Health purchased a third linear accelerator that is expected to be installed soon. Training of hospital staff is greatly improved. A foundation led by a prominent Panamanian cancer survivor, Marta Estela C. de Vallarino, has raised hundreds of thousands of dollars that have helped the hospital buy new mammography and endoscopy machines.

Then there is the restorative power of family. Garcia survived because, after six treatments at the National Cancer Institute, he was so sick that his six children chipped in the $1,500 it cost to finish his treatments at a private hospital.

At that hospital he was treated by, among others, Saldaña, who moonlights there on a second shift.

CODE OF HONOR

—By John McCormick

It's time for a change or two. Or six.

Fundamental problems with the way organizations develop software go, if not ignored, largely unaddressed for far too long. Instead of refusing to employ flawed software, buyers accept bugs, vulnerabilities, corrupt files, system crashes and unpredictable behavior as a cost of business. Weak programming practices mean not just infections of code, but, in the worst cases, revenue- and profit-sapping downtime for corporations, and injuries or even fatalities for humans.

This isn't to say that some software quality isn't high. Safety-minded, serious developers have built systems that allow remote-control vehicles to roam the dusty soil of Mars, let telescopes peer through the vastness of space to glimpse the universe's distant past, and permit jet fighters to stealthily pierce the sky.

Yet everyday life now can't run without reliable software: in appliances, tools and toys; in pacemakers, infusion pumps and radiation-therapy machines; in factories, power plants and office campuses; in trains, planes and automobiles.

Precisely because of software's ubiquity–especially in the machines entrusted with people's lives–"good" is no longer good enough. Only rock-solid software that users can operate without fail and that machines can follow predictably is permissible now.

"There's a huge amount to be done," says James Gosling, the Sun Microsystems vice president who was instrumental in the development of the Java software-development product line.

Where to begin?

Baseline gathered the opinions of more than 20 software and safety experts—including Gosling; Bill Joy, former chief scientist at Sun; Herb Krasner, director of the Software Quality Institute (SQI) at the University of Texas; William Guttman, director of the Sustainable Computing Consortium (SCC); Mike Konrad, a senior member of the Software Engineering Institute (SEI); Pradeep Khosla, who heads the Electrical and Computer Engineering department at Carnegie Mellon; Gary McGraw, chief technology officer at Cigital; and Adam Kolawa, chief executive of Parasoft.

Here's their collective prescription for fixing what ails software development.

1 CERTIFY PROGRAMMERS.

Too many people building programs lack skills. "Lots of people call themselves software engineers who are not," says the SQI's Krasner.

This often means the original design specifications for a software product are inadequate. In the end, these "engineers" can't assess the risk that the software may not work as intended.

To be a doctor, one must get a college degree, pass medical exams, complete an internships and than take a series of tests to practice in a particular specialty. Accountants, engineers and lawyers also most go through rigorous testing and certification processes.

"That doesn't happen in software," Cigital's McGraw says. "You can declare yourself a software architect and off you go."

Organizations such as the Institute for Certification of Computing Professionals (ICCP), the Institute of Electrical and Electronics Engineers (IEEE) and the SEI give exams that cover everything from systems development to data management to the various tools and techniques being used by developers.

But, in the end, it's the companies paying for software that hold the power to demand certification. Today, too few even consider whether the software they buy comes from certified developers.

2 CERTIFY TEAMS.

Software creation is increasingly a collaborative process. That has led to systematic approaches of reviewing the quality of team-created applications. The Capability Maturity Model, developed by the Software Engineering Institute at the request of the Defense Department, establishes whether a given organization has mastered good software development practices. These include the reliable setting of specifications; proper means of evaluating code that has been created; the ability to set and track internal performance metrics; and consistently finding better ways to develop software. Organizations work their way up five levels of maturity model team certification.

Parasoft's Kolawa says a software professional also ought to be certified in a particular industry, be it finance, pharmaceuticals or aerospace. If software quality is going to take any leap forward, Kolawa says, "this type of certification of specialty will have to happen."

3 CHECK, RECHECK AND CHECK AGAIN.

As exemplified by the unexpected fatalities that resulted from the way radiation machines were used in Panama, developers never can anticipate fully how their applications will be used. Yet too often developers don't spend the time and energy needed to find out what users really want and need.

McGraw calls this the "sneaky, dirty little secret of software development." Even in conventional business applications, the developer philosophy often is: "If they don't tell us exactly what they want, we'll just give them something," he says.

The starting point for fixing this software-development flaw is simple: a precise list of what a new program is supposed to do that can be agreed upon by the people developing the software *and* the people who will use it. Then teams must double- and triple-check the code they create to make sure users can't ask it to do tasks that aren't anticipated or cause unexpected conflicts in calculations.

4 RAISE THE BAR.

Building software requires engineering as serious as the kind required for high-rise offices. Just as there are building codes for skyscrapers, so now are serious developers following code codes.

In its Code Conventions for the Java programming language, Java's progenitor, Sun Microsystems, clearly delineates the number of code statements—known as "declarations"—that should be written per screen line (one) and how long each line should be (not more than 80 characters). These conventions recognize such basic facts of code-writing life as how computer terminals "wrap" lines of characters that appear on their screens. The conventions also outline how to clearly name files and how to insert helpful comments into lines of code.

Code conventions are important because they make the code easier to read—and maintain—by people who haven't worked on it.

Jack Ganssle, an engineer whose Ganssle Group advises companies and developers on how to create high-quality software, acknowledges that "a lot of software engineers think that this [discipline] is totally worthless—a way to depress their creativity."

But, he notes, "they're wrong. If the source code is not readable, [if] it's not absolutely clear, how do you think you can possibly audit it, understand what it's doing and look for errors?"

5 TEST, RETEST—AND ESTABLISH A SEAL OF SAFETY.

Sure, a team can test its code and still not find all the problems. But too often, that observation is used as a reason to avoid further testing, not just in development but after the code's put into use.

Any given program can be tested for reliability, security and performance when it's completed. But software can be tested even when it is just a "component" of a system.

Testing tools are widely available from such firms as Empirix, Mercury Interactive, Parasoft and Software Development Technologies (SDT). But, says Gosling, "people don't use them."

Testing ties up personnel, and adds to a project's overall cost. Since many organizations wait until the end of development to test, projects that are just about to come in "on time" and "within budget" often fail to do either.

Krasner, Guttman, Gosling and others agree that one solution can be a software version of the independent, not-for-profit Underwriters Laboratory, which reviews electronic equipment. Such an independent service would provide a seal of approval that a given piece of software or a software-based system is safe. Vendors who find safety to be a fundamental feature of their product—those whose software runs equipment that affects human lives, for instance—would voluntarily submit their products to the lab. If the software checked out as safe and reliable, it would be stamped as suitable for life-critical applications.

Independent testing isn't exactly new. For more than two decades, the National Software Testing Lab in Blue Bell, Pa., for instance, has been creating and managing tests for everything from servers to wireless devices to software applications. Its clients include Dell, Intel, Nokia and the Canadian government. Keylabs of Linton, Utah, which says it has done work for American Airlines, Charles Schwab and Visa, provides similar services.

But there's no generally accepted seal of approval for software.

6 DON'T BUY PROBLEMS.

Perhaps the biggest reason mediocre software persists—and threatens lives—is that individuals and corporations keep buying it.

"People put up with it," says Jonathan Jacky, a scientist working at Microsoft Research.

Software might be the only product designed by a group of people called engineers that's released and known to be imperfect. No one expects a building to fall, a bridge to collapse, a train to derail or a plane to crash. When any of those fail, shock is followed by accusations, inquiries, penalties, and, sometimes, legislative efforts to make sure the problem doesn't recur.

Not so with software. According to the Cutter Consortium, an information-technology consultancy, almost 40% of 150 software-development organizations it polled last year said they didn't believe their organizations had an adequate program in place to ensure that their software was high quality.

Cutter senior consultant Elli Bennatan notes that 29% said their companies didn't have a quality-assurance professional on staff with any real authority, 27% said their companies didn't conduct formal quality reviews, and 24% didn't bother to collect software-quality metrics.

And 32% said their companies released software with too many defects.

"If you don't demand quality, you don't get it," SQI's Krasner says.

In effect, users and developers of software must begin demanding quality, and backing those organizations that certify developers, such as SEI, or those that support development of reliable code, such as the SCC.

Otherwise, it will be lawyers of victims, like those in Panama, and legislators or regulators that will be demanding it—in civil court and in statehouses. Or, in the worst case, in the penal code.

TAKING ACTION

To find out more about the Sustainable Computing Consortium, including how to join the organization, contact Larry Maccherone, Associate Director, CyLab, (412) 268-1715; LMaccherone@cmu.edu

To find out more about the Software Engineering Institute and the Capability Maturity Model, go to www.sei.cmu.edu/cmmi/ or e-mail customer-relations@sei.cmu.edu

For information on ICCP certification, go to www.iccp.org

For information on IEEE certification, go to www.computer.org/certification/