

4 Can We Trust the Computer?

4.1 What Can Go Wrong?

4.1.1 Questions About Reliability and Safety

Most computer applications, from consumer software to systems that control airplanes and telephone networks, are so complex that it is virtually impossible to produce a program with no errors.

Are computer-controlled medical devices, factory automation systems, and airplanes too unsafe to use?

Computer glitches and system failures also have myriad causes, including faulty design, sloppy implementation, careless or insufficiently trained users, and poor user interfaces.

How much risk must or should we accept? If the inherent complexity of computer systems means they will not be perfect, how can we distinguish between errors to be accepted as trade-offs for the benefits of the system and errors that are due to inexcusable carelessness, incompetence, or dishonesty?

We play several roles that allow us to better understand computer-related problems:

- *A computer user:* We must recognize that, as in other areas, there are good products and bad products.
- *A computer professional:* Studying computer failures should help you become a better computer professional.
- *An educated member of society:* There are many personal decisions and social, legal, and political decisions that depend on our understanding of the risks of computer system failures.

4.1.2 Problems for Individuals

BILLING ERRORS

Errors to examine whose negative consequences were relatively easily undone.

- A woman was billed \$6.3 million for electricity; the correct amount was \$63. The cause was an input error.
- In 1998, the IRS sent 3000 people bills for slightly more than \$300 million.
- The auto insurance rate of a 101-year-old man suddenly tripled. The program was written to handle ages only up to 100. It mistakenly classified the man as a teenager.

When mistakes are as big as these, they are obvious, and the bills are corrected.

DATABASE ACCURACY PROBLEMS

Like \$40-billion tax bills, credit-record errors that result from a systematic error affecting thousands of people are likely to be recognized and corrected. More serious perhaps are all the small errors in individual people's records.

A county agency used the wrong middle name in a report to a credit bureau about a father who did not make his child-support payments. Another man in the same county had the exact name reported; he could not get credit to buy a car or a house.

Studies of the FBI's National Crime Information Center (NCIC) database in the 1980s found that roughly 11% of the arrest warrants listed in it were inaccurate or no longer valid.

A Michigan man was arrested for several crimes, including murders, committed in Los Angeles.

Another man had assumed his identity after finding his lost wallet.

A college professor returning from London was arrested and jailed for two days after a routine check with NCIC at Customs showed that he was a wanted fugitive. NCIC was wrong—for the third time about this particular man.

The FBI maintains another database, the Interstate Identification Index (III), containing arrest and conviction files on major offenders from participating states and available to police throughout most of the country. In some states, it is also used by licensing boards and employers.

Several factors contribute to the frequency and severity of the problems people suffer because of errors in databases:

- A large population (Many people have identical or similar names, and most of our interactions are with strangers)
- Automated processing without human common sense or the power to recognize special cases
- Overconfidence in the accuracy of data stored on computers
- Errors (some due to carelessness) in data entry
- Failure to update information and correct errors
- Lack of accountability for errors.

4.1.3 System Failures

Modern communications, power, medical, financial, retail, and transportation systems depend heavily on computers. The computers do not always function as planned.

COMMUNICATIONS

Nationwide AT&T telephone service for voice and data was disrupted for nine hours in January 1990 because of a software error in a four-million line program.

In 1996, Bell Atlantic's directory assistance system failed when an upgrade was installed.

Satellites are a fundamental part of our communications systems. When a Galaxy IV satellite computer failed in 1998, many systems we take for granted stopped working. Pager service stopped for an estimated 85% of users in the U.S., including hospitals and police departments.

BUSINESS, FINANCIAL, AND TRANSPORTATION SYSTEMS

In 2000, a computer malfunction froze the London Stock Exchange for almost eight hours.

A failure of Amtrak's reservation and ticketing system during Thanksgiving weekend in 1996 caused delays because agents had no printed schedules or fare lists.

American Express Company's credit card verification system failed during the Christmas shopping season in 1999.

DESTROYING BUSINESSES

A few dozen companies that bought an inventory system called Warehouse Manager blamed the system for disastrous losses. The program was sold by NCR Corporation, but it was developed by another company. It was originally designed for and implemented on a different computer and operating system.

STALLED AIRPORTS: DENVER, HONG KONG, AND MALAYSIA

Denver International Airport: There were no airplanes or people at the airport and no cars on the highway—10 months after the \$3.2 billion airport was to have opened.

Most of the delay was attributed to the computer-controlled baggage-handling system, which cost \$193 million.

ABANDONED SYSTEMS

The California Department of Motor Vehicles, for example, abandoned a \$44 million computer system that never worked properly. A consortium of hotels and a rental car business spent \$125 million on a comprehensive travel-industry reservation system, then canceled the project because it did not work.

After spending \$4 billion, the IRS abandoned a tax-system modernization plan; a General Accounting Office report blamed mismanagement.

4.1.4 Safety-Critical Applications

COMPUTERS IN THE AIR

The A320 Airbus airplane was the first fully “fly-by-wire” airplane.

Between 1988 and 1993, four A320s crashed. Although the official cause for some of the crashes was ruled “pilot error,” pilots and some observers blamed the fly-by-wire system.

The Traffic Collision Avoidance System (TCAS) detects a potential in-air collision and directs the airplanes to avoid each other.

The first version of the system had so many false alarms that it was unusable.

4.2 Case Study: The Therac-25

4.2.1 Therac-25 Radiation Overdoses

The Therac-25 was a software-controlled radiation-therapy machine used to treat people with cancer. Between 1985 and 1987, Therac-25 machines at four medical centers gave massive overdoses of radiation to six patients. In some cases, the operator repeated an overdose because the machine’s display said that no dose had been given. Medical personnel later estimated that some patients received between 13,000 and 25,000 rads (a rad is the unit used to quantify radiation doses. It stands for “radiation absorbed dose.”), where the intended dose was in the 100-200 range. These incidents caused severe and painful injuries and the deaths of three patients.

The computer monitors and controls movement of a turntable on which three sets of devices are mounted. Depending on whether the treatment is electron or X-ray.

A third position of the turntable may be used with the electron beam off, and a light beam on instead, to help the operator position the beam in precisely the correct place on the patient’s body.

4.2.2 Software and Design Problems

DESIGN FLAWS

The Therac-25, developed in the late 1970s, followed earlier machines called the Therac-6 and Therac-20.

It was designed to be fully computer controlled. The older machines had hardware safety interlock mechanisms.

Many of these hardware safety features were eliminated in the design of the Therac-25. Some software from the Therac-20 and Therac-6 was reused in the Therac-25.

The Therac-20 software had bugs, but the hardware safety mechanisms were doing their job.

The Therac-25 malfunctioned frequently.

Operators became used to error messages appearing often, with no indication that there might be safety hazards.

The error messages that appeared on the display were simply error numbers or obscure messages (“Malfunction 54” or “H-tilt”).

The operator’s manual for the Therac-25, however, did not include any explanation of the error messages. Even the maintenance manual did not explain them.

BUGS

Investigators were able to trace some of the overdoses to two specific software errors.

The Set-Up Test procedure can run several hundred times while setting up for one treatment. A flag variable is used to indicate whether a specific device on the machine is positioned correctly. A zero value means the device is ready.

Each time the Set-Up Test procedure runs, it increments the variable to make it nonzero. The problem was that the flag variable was stored in one byte. When the routine was called the 256th time, the flag overflowed and showed a value of zero.

This bug allowed the electron beam to be turned on when the turntable was positioned for use of the light beam, and there was no protective device in place to attenuate the beam.

Part of the tragedy in this case is that the error was such a simple one, with a simple correction. No good student programmer should have made this error. The solution is to set the flag variable to a fixed value, say 1, rather than incrementing it, to indicate that it must be checked.

4.2.3 Why So Many Incidents?

There were six known Therac-25 overdoses. You may wonder why the machine continued to be used after the first one.

It was not immediately pulled from service after the first few accidents because it was not known immediately that it was the cause of the injuries.

The manufacturer was questioned about the possibility of overdoses, but responded (after the first, third, and fourth accidents) that the patient injuries could not have been caused by the machine.

After the fifth accident, the FDA declared the machine defective and ordered AECL to inform users of the problems.

OVERCONFIDENCE

In the first overdose incident, when the patient told the machined operator that she had been “burned,” the operator told her that was impossible.

The most obvious and critical indication of overconfidence in the software was the decision to eliminate the hardware safety mechanisms.

They did not expect significant problems from software errors.

4.2.4 Observations and Perspective

From design decisions all the way to responding to the overdose accidents, the manufacturer of the Therac-25 did a poor job.

The number and pattern of problems in this case, and the way they were handled, suggest serious irresponsibility.

They suggest, however, that individual and management responsibility, good training, and accountability are factors more important than whether or not a computer is used.

4.3 Increasing Reliability and Safety

4.3.1 What Goes Wrong?

Computer systems now interact with the real world (including both machinery and unpredictable humans) have numerous features and interconnected subsystems, and are extremely large. Computer programs have tens of thousands, hundreds of thousands, or millions of lines of code. (Microsoft's Windows 2000 has about 40 million lines.) Computer software is "non-linear" in the sense that, whereas a small error in an engineering project may cause a small degradation in performance, a single typo in a computer program can cause a dramatic difference in behavior.

Some factors in computer system errors and failures:

- Interaction with physical devices that do not work as expected.
- Incompatibility of software and hardware, or of application software and the operating system.
- Management problems, including business and/or political pressure to get a product out quickly.
- Inadequate attention to potential safety risks.
- Not planning and designing for unexpected inputs or circumstances.
- Insufficient testing
- Reuse of software from another system without adequate checking.
- Overconfidence in software.
- Carelessness.
- Misrepresentation; hiding problems; inadequate response when problems are reported.
- Problems with management of the use of a system:
 - Data-entry errors.
 - Inadequate training of users.
 - Errors in interpreting results or output.
 - Overconfidence in software by users.
 - Insufficient planning for failures; no backup systems or procedures.
- Lack of market or legal incentives to do a better job.

OVERCONFIDENCE

Even when programmers work separately, they tend to make the same kinds of errors, especially if there is an error, ambiguity, or omission in the program specifications.

Political pressure to produce inflated safety predictions is not restricted to computer systems.

REUSE OF SOFTWARE: THE ARIANE 5 ROCKET

Less than 40 seconds after the first Ariane 5 rocket was launched in 1996, it veered off course and was destroyed as a safety precaution.

The Ariane 5 used some software designed for the earlier, successful Ariane 4. The Ariane 5 travels faster than the Ariane 4 after takeoff. The calculations produced numbers bigger than the program was designed to handle (an “overflow” in the technical jargon), causing the system to halt.

Especially for safety-critical applications, it is essential to reexamine the specifications and design of the software, and to retest it, when it is to be used in a new environment.

4.3.2 Professional Techniques

SOFTWARE ENGINEERING AND PROFESSIONAL RESPONSIBILITY

Most accidents are not the result of unknown scientific principles but rather of a failure to apply well-known, standard engineering practices.

Accidents will not be prevented by technological fixes alone, but will require control of all aspects of the development and operation of the system.

The risks of turning control over to computers must be weighed carefully. Most software today is simply not safe enough for safety-critical applications. Hardware safety mechanisms, as used by engineers in pre-computer systems, still have an important role.

USER INTERFACES AND HUMAN FACTORS

Well-designed user interfaces can help avoid many computer-related problems.

User interfaces should provide clear instructions and error messages; they should be consistent; and they should include appropriate checking of input to reduce major system failures caused by typos or other errors a person can be reasonably expected to make.

Good user interfaces are essential in safety-critical applications.

REDUNDANCY AND SELF-CHECKING

AT&T's telephone system handles roughly 100 million calls a day. The system is designed to constantly monitor itself and correct problems automatically. Half of the computing power of the system is devoted to checking the rest for errors.

Even when the best professional practices are followed, even with extensive testing, we cannot be guaranteed that such complex systems do not have bugs.

TESTING

It is difficult to overemphasize the importance of adequate, well-planned testing of software. Unfortunately, many programmers and software developers see testing as a dispensable luxury, a step to be skimmed on to meet a deadline or to save money.

4.3.3 Law and Regulation

CRIMINAL AND CIVIL PENALTIES

Legal remedies for faulty systems include suits against the company that developed or sold the system and criminal charges when fraud or criminal negligence occurs.

Many contracts for business computer systems limit the amount the customer can recover to the actual amount spent on the computer system.

Well-designed liability laws and criminal laws—not so extreme that they discourage innovation, but clear and strong enough to provide incentives to produce good systems—are important legal tools for increasing reliability and safety of computer systems.

WARRANTIES FOR CONSUMER SOFTWARE

Most mass, retail consumer software, from word processors to games are sold “as-is;” there is no guarantee that they work correctly.

Consumers are paying for a product that works, and fairness dictates that they get one that does.

REGULATION AND SAFETY-CRITICAL APPLICATIONS

If the FDA had thoroughly examined the Therac-25 before it was put into operation, the flaws might have been found before any patients were injured.

The issues involved in regulation of risky technology are complex. Overly strict standards can inhibit progress, require techniques behind the state of the art, and transfer responsibility from the manufacturer to the government. The fixing of responsibility requires a delicate balance.

Someone must represent the public's needs, which may be subsumed by a company's desire for profits. On the other hand, standards can have the undesirable effect of limiting the safety efforts and investment of companies that feel their legal and moral responsibilities are fulfilled if they follow the standards. Some of the most effective standards and efforts for safety come from users. Manufacturers have more incentive to satisfy customers than to satisfy government agencies.

PROFESSIONAL LICENSING

Another controversial approach to improving software quality is mandatory licensing of software development professionals. The desired effect is to protect the public from poor quality and unethical behavior.

4.3.4 Taking Responsibility

Intuit offered to pay interest and penalties that resulted from errors in flawed income-tax programs.

We noted that business pressures are often a reason for cutting corners and releasing defective products. Business pressure can also be a cause for insistence on quality and maintaining good customer relations.

4.4 Perspectives on Failures, Dependence, Risk, and Progress

4.4.1 Putting Failures in Perspective

How accurate should software for check processing be? 99%? 99.9%?

At some point, the expense of improving a system is not worth the gain, especially for applications where errors can be detected and corrected at lower cost than it would take to try to eliminate them.

COMPLEX SYSTEMS

On October 28, 1997, 1.2 billion shares of stock were traded on the New York Stock Exchange—76% more than the previous record. The Stock Exchange computers handled the sales without errors or delays. The Exchange managers had planned in advance, spending \$2 billion on a system with 450 refrigerator-size computers, 200 miles of fiberoptic cable, 8000 telephone circuits, and 300 data routers.

Many large, complex, expensive computer systems work.

The new GWPS is credited with helping avoid a crash into a mountain by an Air France plane attempting to land in conditions of poor visibility.

There is controversy about how much control should be given to a flight computer. Based on accident statistics, some airlines agree that more lives can be saved by preventing pilots from doing something “stupid” than by letting them do something outside the program limitations in the rare cases where that might be needed.

4.4.2 Are We Too Dependent On Computers?

A fire at a telephone switching facility in Los Angeles disrupted telephone service for half a day.

The incident serves as a good reminder about how many ordinary daily activities are dependent on communications and computer networks.

When an AT&T system used by banks failed, a supermarket manager reported “Customers are yelling and screaming because they can’t get their money, and they can’t use the ATM to pay for groceries.”

Because of their usefulness and flexibility, computers are now virtually everywhere. Is this good? Or bad? Or neutral?

“DEPENDENCE” OR “USE”?

Is our “dependence” on computers different from our dependence on electricity? Is it different from a farmer’s dependence on a plow?

When we have a good tool, we can forget, or no longer even learn, the older method of performing a task. If the tool breaks down, we are stuck; we cannot perform the task until the tool is fixed.

The negative effects of a breakdown do not condemn the tool.

The inconveniences or dangers of a breakdown are a reminder of the convenience and productivity provided by the tool when it is working, for example, of the billions of telephone calls (carrying voice, e-mail, files, and data) that are completed—that are made possible or more convenient or cheaper because of computers.

RISK AND PROGRESS

Most new technologies were not very safe when they were first developed. If the death rate from commercial airline accidents in the U.S. were the same now as it was 50 years ago, 8,000 people would die in plane crashes each year (instead of fewer than 200).

The American Airlines plane that crashed near Cali, Columbia is used as an example in software engineering textbooks, so that future software specialists will not repeat the mistakes in the plane’s computer system. We learn.

As use of technology, automation, and computer systems has increased in virtually all work places, the risk of dying on an on-the-job accident dropped from 390 in one million (in 1934) to 40 in one million (in 1994).

Safety: mistakes made in software are the same as those that used to be common in engineering.

There are some important differences between computers and other technologies. Computers make decisions; electricity does not. The power and flexibility of computers encourages us to build more complex systems—where failures have more serious consequences. The pace of change in computer technology is much higher than that in other technologies. Software is not built from standard, trusted parts as is the case in many engineering fields. The software industry is still going through its “growing pains;” it has not yet developed into a mature, fully developed discipline.

FALSE CHARGES—OR, BLAMING THE STOVE FOR A POORLY COOKED MEAL

The body of a reclusive, elderly man who died in his apartment was not discovered until six months after his death.

Many of the man’s bills, including rent and utilities, were paid automatically, and his pension check was automatically deposited in his bank account. Thus “all the relevant authorities assumed that he was still alive.”

4.4.3 Observations

Several important points:

1. Many of the issues related to reliability for computers have arisen before with other technologies.
2. Perfection is not an option. The complexity of computer systems makes errors, oversights, and so on likely.
3. There is a “learning curve” for new technologies. By studying failures, we can reduce their occurrence.
4. Risks of using computers should be compared with risks of other methods and with benefits obtained.

This does not mean that computer errors and failures should be excused or ignored because failures occur in other technologies. It does not mean that carelessness or negligence should be tolerated because perfection is not possible. It does not mean that accidents should be excused as part of the learning process, and it does not mean that accidents should be excused because, on balance, the contribution of computers is positive.

4.5 Computer Models

4.5.1 Evaluating Models

Some problems studied with computer models:

- Population growth.
- The cost of a proposed government program.
- The effects of second-hand smoke.
- When we will run out of a critical natural resource.
- The effects of a tax cut on the economy.
- The threat of global warming.
- When a big earthquake is likely to occur.

Computers are used extensively to model and simulate both physical systems, such as the design for a new car or the flow of water in a river, and intangible systems, such as parts of the economy.

They have obvious social and economic benefits: They help train operators of power plants, submarines, and airplanes.

Predictions from expensive computers and complex computer programs impress people, but models vary enormously in quality.

It is the professional and ethical responsibility of those who design and develop models for public issues to describe honestly and accurately the results, assumptions, and limitations of their models.

Are reusable (washable cloth) diapers better for the environment than disposable diapers?

To illustrate how difficult such a study may be, below is a list of the questions about which the modelers made assumptions. Depending on the assumptions, the conclusions differed.

- How many times is a cloth diaper used before it is discarded? (Values ranged from 90 to 167).
- Should credit be given for energy recovered when waste is incinerated, or does pollution from incineration counterbalance the benefit?
- What value should be assigned for the labor cost of washing diapers?
- How many cloth diapers are used each time a baby is changed? (Many parents use two at once for increased protection.) The models used values ranging from 1.72 to 1.9.
- How should the pesticides used in growing cotton be counted?

The following questions help us determine the validity and accuracy of a model:

1. How well do the modelers understand the underlying science or theory (be it physics, chemistry, economics, or whatever) or a system being studied? How well understood are the relevant properties of the materials involved? How accurate and complete are the data?
2. Models necessarily involve assumptions and simplifications of reality. What are the assumptions and simplifications in the model?
3. How closely do the results or predictions of the model correspond with results from physical experiments or real experience?

4.5.2 Car Crash-Analysis Programs

4.5.3 Climate Models and Global Warming

CONCLUSIONS AND ETHICAL ISSUES

The models are a tool for understanding climate change, but are not yet at a stage where we can rely on the precision of their results. Certainly they have not achieved the level of reliability of the car-crash-analysis models.

Scientific papers about climate models routinely describe the uncertainties and limitations of their results. News headlines, and even the policy summaries provided with the IPCC reports, sometimes do not.