

# CS 107 Fall 2006

## Chapter 12: Issues and Ethics

### 1 Why Things Don't Work

Why don't programs do what we want, need and expect?

1. Tiny hardware failures can have far-reaching consequences (NORAD multiplexer incident)
2. Programmers sometimes use the wrong formula (space capsule landing 100 miles from the target)
3. People cannot anticipate every possible kind of input data (rising moon interpreted as missile attack)
4. Human stupidity (Chernobyl, operators shut down the safety systems for testing)
5. Cutting corners to get high speed computations (Ariane crash on take-off)
6. Cutting corners to get the product to market faster. (Windows)
7. Design for flashiness or sex-appeal without regard for security, stability, quality. (Windows)
8. Complexity: with a very large system, NOBODY understands all of it.
9. Concurrency: two processes using the same data can turn good data into garbage.

We will discuss these problems one at a time.

**1. Hardware failure.** Adding redundant hardware and software does not fix the reliability problem. Why? That adds complexity, and then you need tie-breaking software, which adds complexity. Complexity is the #1 big reason that programs do not work.

**2. Miscommunication or missing communication.** Programmers often misunderstand the client's needs. There is always pressure to get the work done fast and cheaply. Time spent communicating is sometimes viewed as wasted time.

**3. Abstractions eliminate detail.** Computer data is an abstraction. What happens when two different physical events result in the same abstraction? Things get mixed up.

**4. Operators are human.** There will never be an end to human error. We can try to design systems to be safe even when humans make errors, but if the operator turns off the safety system, then what?

**5. Faster and faster.** Engineers have bosses. The bosses understand computers and computation even less than the engineers, but they are the bosses. If they say "jump", the engineers have to jump or prove it can't be done. Bosses will always push to make code go faster. Taking shortcuts seems like the easy way out. (This is not responsible, but it doesn't make waves.) When this causes the programmer to skip making routine validity checks, things can go very badly wrong.

For example, the unmanned Ariane 5 rocket, developed by the European space agency exploded during launching on its first voyage. The cause was failure to test for and catch the result of a computation that was too big for the computer to store. The wrong number was stored and the rocket went off course and exploded. This was a \$500 million disaster. The programmer omitted the test for this kind of an error because he was trying to make the program run faster and he believed that the problem could never arise.

**6. Market pressure.** There will always be a push to get the product to market sooner. The second to finish might be better but it won't make as much money. (Windows vs. OS-X) Unfortunately, this encourages people to take design shortcuts and skimp on testing.

**7. The marketing department.** Security does not sell products; flashiness does. Windows has been developed by people who believe that new features, animation, pop-ups, and bright colors will sell systems. They said that nobody cares about security. And of course, everyone EXPECTS a computer system to lock up or crash all the time. Unfortunately, much of the public accepts these things, possibly out of ignorance.

**8. Complexity.** Many systems are unbelievably huge – so large that nobody understands the entire system. Building such a system requires a disciplined approach to making parts that fit smoothly together and work properly. Software engineering best-practices are our best defense against errors due to complexity. However, much or maybe most applications are not built following these practices. Think about SAM.

**9. Concurrency.** Concurrent systems such as online databases must implement controls over concurrent use of the same data. Commercial databases such as Oracle provide good support for concurrency control by locking data records when one process uses them. Other kinds of applications often don't provide this kind of support.

## 2 Basic Rights

**Freedom of Speech.** Do we have the legal right to censor hate sites, terrorist sites, bomb making sites, etc. Should parents use programs such as Net Nanny to protect their kids ?

A *hactivist* is a hacker who attacks undesirable sites that carry porn, hate literature, etc. They also have attacked companies and government sites they disapprove of. Should this illegal activity be prosecuted? Or should we encourage these kids?

**Privacy.** We have two tools, encryption and https, to combat snooping and unintentional theft of data. Unfortunately, they are not always used when they should be.

When you post personal information on myspace, you expose yourself to identity theft, stalking, and snooping by employers. Don't post compromising or identifying facts. Clean off your "wall" regularly (or take yourself off myspace).

**Surveillance.** Many kinds of surveillance are possible, including

- A company can monitor email and web usage if they have a specific policy on those things.
- The FBI monitors the traffic of known and suspected terrorists.
- Cell phone calls can be located using a GPS. Cars can be located if they have an embedded marker. People have used embedded markers for pets and proposed them for children.
- We could have national ID cards with an RFID chip. There are current proposals to put these chips in all driver's licenses.

### 3 Other Social Issues

1. The *digital divide*. Some kinds of users do not have equal access to telecommunications. This includes some disabled users and the poor. Currently there is a movement to develop a \$100 laptop and Ubuntu Linux (free, easy to use) to meet the needs of more of the poor. Equalizing access would also require providing access to broadband networks for everybody.
2. The internet is used for many purposes that are damaging to society: hacking, zombies, terrorism, phishing, scams, kiddie porn. Can and should these things be regulated?
3. Is there anything we can do about unwanted but legal intrusion: spam, hoaxes, chain letters, spyware, adware, and cookies. <http://hoaxbusters.ciac.org/>
4. Legal porn and gambling sites often lead to addiction. Should they be regulated? Illegal? Do we need programs like *Net Nanny* to block access by children? Is there a better way?
5. Health problems:
  - Internet addiction: It becomes a mental-health problem when browsing and gaming displace sleep, homework, exercise, and relating to people. Worse, a lonely soul who discovers an inviting online community can be drawn into illegal activity such as terrorism or hacking.
  - Carpal tunnel syndrome can be the result of heavy daily use plus a poor arrangement of the computer workstation, keyboard, and chair.
  - Eye strain can result from small type fonts, low-quality monitors, and poor setup.