CSE/ISE 312

Chapter 8: Errors, Failures, and Risk

What We Will Cover

- Failures and Errors in Computer Systems
- Case Study: The Therac-25
- Increasing Reliability and Safety
- Dependence, Risk, and Progress

Failures and Errors in Computer Systems

- Most computer applications are so complex it is virtually impossible to produce programs with no errors
- The cause of failure is often more than one factor
  - Faulty design, sloppy implementation, careless users, poor user interface, insufficient user training…
- Design and testing of mission critical systems is much more complex than typical computer-based systems
- Computer professionals must study failures to learn how to avoid them, and to understand the impacts of poor work

Problems for Individuals

- Billing errors
- Inaccurate and misinterpreted data in databases
  - Large population where people may share names
  - Automated processing may not be able to recognize special cases
  - Overconfidence in the accuracy of data
  - Errors in data entry
  - Lack of accountability for errors

System Failures

- AT&T, Galaxy IV satellite, Amtrak
- Businesses have gone bankrupt after spending huge amounts on computer systems that failed
- Voting systems in presidential elections
- Stalled airports: Denver, Hong Kong, Malaysia
- Abandoned systems
  - Systems discarded after wasting millions even billions of dollars
- Legacy systems
  - Reliable but inflexible, expensive to replace, little documentation

Denver Airport

- Baggage handling system costs ~ $200 million, caused most of the delay
- Baggage system failed due to real world problems, problems in other systems and software errors
  - Carts crashed into each other at track intersections, mistaken route. Scanner got dirty or knocked out of alignment, faulty latches, power surges
- Main causes:
  - Time allowed for development was insufficient
  - Denver made significant changes in specifications after the project began

High-level, management-related causes of computer-system failures

- Lack of clear, well thought out goals and specifications
- Poor management decisions and poor communication among customers, designers, programmers, etc.
- Institutional and political pressures that encourage unrealistically low bids, low budget requests, and underestimates of time requirements
- Use of very new technology, with unknown reliability and problems
- Refusal to recognize or admit a project is in trouble

Case Study: The Therac-25

Therac-25 Radiation Overdoses:

- Therac-25: a software controlled radiation therapy machine used to treat cancer patients
- 1985-1987, 4 medical centers
- Massive overdoses of radiation were given; the machine said no dose had been administered at all
- Caused severe and painful injuries and the death of three patients
- Important to study to avoid repeating errors
- Manufacturer, computer programmer, and hospitals/clinics all have some responsibility

Case Study: The Therac-25 (cont.)

Software and Design problems:

- Re-used software from older systems, unaware of bugs in previous software
- Weaknesses in design of operator interface
  - Obscure error messages with no documentation on them
- Inadequate test plan
- Bugs in software
  - Allowed beam to deploy when table not in proper position
  - Ignored changes and corrections operators made at console

Therac-25 - Why So Many Incidents?

- Hospitals had never seen such massive overdoses before, were unsure of the cause
- Manufacturer said the machine could not have caused the overdoses and no other incidents had been reported (which was untrue)
- The manufacturer made changes to the turntable and claimed they had improved safety after the second accident. The changes did not correct any of the causes identified later

Therac-25 - Why So Many Incidents?

- Recommendations were made for further changes to enhance safety; the manufacturer did not implement them
- The FDA declared the machine defective after the fifth accident
- The sixth accident occurred while the FDA was negotiating with the manufacturer on what changes were needed

Observations and Perspective

- Minor design and implementation errors usually occur in complex systems; they are to be expected
- The problems in the Therac-25 case were not minor and suggest irresponsibility
- Accidents occurred on other radiation treatment equipment without computer controls when the technicians:
  - Left a patient after treatment started to attend a party
  - Did not properly measure the radioactive drugs
  - Confused micro-curies and milli-curies

Increasing Reliability and Safety

What goes wrong?

- Design and development problems
- Management and use problems
- Misrepresentation, hiding problems and inadequate response to reported problems
- Insufficient market or legal incentives to do a better job
- Re-use of software without sufficiently understanding the code and without thoroughly testing it
- Failure to update or maintain a database
- Overconfidence

Professional Techniques

- Importance of good software engineering and professional responsibility
- User interfaces and human factors
  - Feedback
  - Should behave as an experienced user expects
  - Workload that is too low can lead to mistakes
- Redundancy and self-checking
- Testing
  - Include real world testing with real users

Law, Regulation and Markets

- Criminal and civil penalties
  - Provide incentives to produce good systems, but shouldn't inhibit innovation
- Warranties for consumer software
  - Most are sold ‘as-is’
- Regulation for safety-critical applications
- Professional licensing
- Taking responsibility

Dependence, Risk, and Progress

- Are We Too Dependent on Computers?
  - Computers are tools
  - They are not the only dependence
    - Electricity
- Risk and Progress
  - Many new technologies were not very safe when they were first developed
  - We develop and improve new technologies in response to accidents and disasters
  - We should compare the risks of using computers with the risks of other methods and the benefits to be gained