

A GIFT OF FIRE

*Social, Legal, and Ethical Issues
for Computing Technology*

Fifth Edition

Chapter 8: Errors, Failures and Risks



SARA BAASE
TIMOTHY M. HENRY

Based on slides prepared by Cyndi Chie,
Sarah Frye and Sharon Gray.

Fifth edition updated by Timothy Henry

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
- Computer professionals must study failures to learn how to avoid them
- Computer professionals must study failures to understand the impacts of poor work



Failures and Errors in Computer Systems

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



Failures and Errors in Computer Systems

System Failures

- Galaxy IV
- Amtrak



Failures and Errors in Computer Systems

System Failures

- Voting systems
 - Technical failures
 - Programmers or hackers rigging software to produce inaccurate results.
 - Vulnerability to viruses



Failures and Errors in Computer Systems

System Failures

- Denver Airport
 - Baggage system failed due to real world problems, problems in other systems and software errors
 - Main causes:
 - Time allowed for development was insufficient
 - Denver made significant changes in specifications after the project began



Failures and Errors in Computer Systems

System Failures

- Airports in Hong Kong and Kuala Lumpur
 - Comprehensive systems failed because designers did not adequately consider potential for user input error.



Failures and Errors in Computer Systems

System Failures

- Abandoned systems
 - Some flaws in systems are so extreme that the systems are discarded after wasting millions, or even billions, of dollars.



Failures and Errors in Computer Systems

System Failures

- Lack of clear, well-thought-out goals and specifications
- Poor management 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



Failures and Errors in Computer Systems

System Failures

- Legacy systems
 - Reliable but inflexible
 - Expensive to replace
 - Little or no documentation



Failures and Errors in Computer Systems

What Goes Wrong?

- The job they are doing is inherently difficult.
- Sometimes the job is done poorly.



Failures and Errors in Computer Systems

What Goes Wrong?

- Design and development problems
 - Inadequate attention to potential safety risks
 - Interaction with physical devices that do not work as expected
 - Incompatibility of software and hardware, or of application software and the operating system
 - Not planning and designing for unexpected inputs or circumstances
 - Confusing user interfaces
 - Insufficient testing
 - Reuse of software from another system without adequate checking
 - Overconfidence in software
 - Carelessness



Failures and Errors in Computer Systems

What Goes Wrong?

- Management and use problems
 - Data-entry errors
 - Inadequate training of users
 - Errors in interpreting results or output
 - Failure to keep information in databases up to date
 - Overconfidence in software by users



Failures and Errors in Computer Systems

What Goes Wrong?

- Misrepresentation, hiding problems and inadequate response to reported problems
- Insufficient market or legal incentives to do a better job



Failures and Errors in Computer Systems

What Goes Wrong?

- Reuse of software: the Ariane 5 rocket and “No Fly” lists
 - It is essential to reexamine the specifications and design of the software, consider implications and risks for the new environment, and retest the software for the new use.



Case Study: The Therac-25

Therac-25 Radiation Overdoses

- 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

Software and Design problems

- Re-used software from older systems, unaware of bugs in previous software
- Weaknesses in design of operator interface
- Inadequate test plan
- Bugs in software
 - Allowed beam to deploy when table not in proper position
 - Ignored changes and corrections operators made at console



Case Study: The 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.

Case Study: The Therac-25

Why So Many Incidents? (cont.)

- 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.

Case Study: The Therac-25

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

Case Study: The Therac-25

Discussion Question

- *If you were a judge who had to assign responsibility in this case, how much responsibility would you assign to the programmer, the manufacturer, and the hospital or clinic using the machine?*

Increasing Reliability and Safety

Professional techniques

- Importance of good software engineering and professional responsibility
- User interfaces and human factors
- Redundancy and self-checking
- Testing
 - Include real world testing with real users

Increasing Reliability and Safety

Professional techniques

- Management and communication
- High reliability organization principles
 - preoccupation with failure
 - loose structure

Increasing Reliability and Safety

Safety-critical applications

- Identify risks and protect against them
- Convincing case for safety
- Avoid complacency



Increasing Reliability and Safety

Specifications

- Learn the needs of the client
- Understand how the client will use the system

Increasing Reliability and Safety

User interfaces and human factors

- User interfaces should:
 - provide clear instructions and error messages
 - be consistent
 - include appropriate checking of input to reduce major system failures caused by typos or other errors a person will likely make

Increasing Reliability and Safety

User interfaces and human factors

- The user needs feedback to understand what the system is doing at any time.
- The system should behave as an experienced user expects.
- A workload that is too low can be dangerous.

Increasing Reliability and Safety

Redundancy and self-checking

- Multiple computers capable of same task; if one fails, another can do the job.
- Voting redundancy

Increasing Reliability and Safety

Testing

- Even small changes need thorough testing
- Independent verification and validation (IV&V)
- Beta testing



Trust the Human or the Computer System?

- Traffic Collision Avoidance System (TCAS)
- Computers in some airplanes prevent certain pilot actions



Law, Regulation, and Markets

- Criminal and civil penalties
 - Provide incentives to produce good systems, but shouldn't inhibit innovation
- Regulation for safety-critical applications
- Professional licensing
 - Arguments for and against
- 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

Dependence, Risk, and Progress

Discussion Questions

- *Do you believe we are too dependent on computers? Why or why not?*
- *In what ways are we safer due to new technologies?*

